



Treball Final de Grau

Maqueta per a l'estudi de la viabilitat d'energies renovables.

**Grau en Enginyeria Electrònica
Industrial i Automàtica**

Curs 15/16

Autor: Víctor Lagén Morancho

Director: Antoni Escobet Canal

Data: 10/07/2016

Localitat: Manresa

Agraïments:

Donar les gràcies al meu tutor Antoni Escobet, ja que sense ell no hagués estat possible la realització d'aquest treball. Sobretot, pel que fa al desenvolupament del programa, m'ha ajudat en tot moment solucionant els meus dubtes, resolent els problemes que anaven sorgint i aportant noves idees per optimitzar el programa en relació a la funcionalitat de la maqueta.

I a la meva família, per donar-me l'oportunitat d'arribar a ser enginyer.

RESUM DEL PROJECTE

Aquest projecte es basa en el disseny d'una maqueta autònoma per ajudar a decidir la viabilitat d'una instal·lació d'energies renovables en una ubicació concreta. El projecte es focalitza en l'estudi de l'energia eòlica i solar.

La maqueta està formada per un sistema enquestat amb un petit aerogenerador i una cel·la solar. Aquesta es dedica a recollir informació dels sensors durant un període de temps.

El projecte està dividit en dues parts. La primera part, i més important, consisteix en l'elaboració pràctica de la part de la maqueta encarregada de recollir informació dels sensors mitjançant dues plataformes; arduino i labwindows. La segona part es basa en l'estudi teòric de proporcionar total autonomia a la maqueta pel que fa a l'alimentació de la placa d'adquisició de dades i a la connexió sense fils de la maqueta amb el PC receptor.

ABSTRACT

This project is based on the design of an autonomous model to help in deciding the feasibility of a renewable energy installation in a particular location. The project is focused on the study of wind and solar power.

The model consists on an embedded system formed by a small wind turbine and a solar cell. Its objective is to collect information from the sensors during a period of time.

The project is divided in two parts. The first one, and most important, is based on the practical elaboration of the model in charge of collecting information from the sensors through two platforms: Arduino and LabWindows. The second one consists on a theoretical study to give totally autonomy to the model in two areas: the alimentation of the data acquisition board and the wireless connection of the model with the receiving computer.

ÍNDEX

1	INTRODUCCIÓ	10
2	ANTECEDENTS	11
3	COS DEL PROJECTE	12
3.1	COMPONENTS	12
3.1.1	Sensors	12
3.1.2	Alimentació	14
3.1.3	Adaptadors	15
3.1.4	Connexió telefònica	23
3.1.5	Arduino	26
3.2	ESTUDI PRÀCTIC MAQUETA	29
3.2.1	Dades Estudi Pràctic	30
3.3	PROGRAMA	31
3.3.1	Elecció Programari Arduino & Programari Ordinador	32
3.3.2	Programa Arduino	33
3.3.3	Programa Labwindows	38
3.4	ESTUDI TEÒRIC PER A LA MAQUETA FINAL	46
3.4.1	Modificacions programa final	48
3.5	VIABILITAT	50
3.5.1	Viabilitat instal·lació solar	50
3.5.2	Viabilitat instal·lació eòlica	52
4	ESTUDI ECONÒMIC	54
5	CONDICIONS PER A LA SEVA EXECUCIÓ	56
5.1	PLANIFICACIÓ UBICACIÓ SENSORS	56
5.1.1	Cobertura GSM	56
5.1.2	Mínim d'hores de sol (per alimentar la bateria)	57
5.1.3	Mínima altura per l'aerogenerador	59

5.1.4	Inclinació i orientació placa solar	59
5.2	MANUAL D'ÚS	60
6	IMPLICACIONS AMBIENTALS	62
6.1	AVANTATGES ENERGIES RENOVABLES	62
6.2	INCONVENIENTS ENERGIES RENOVABLES.....	63
6.2.1	Impacte ambiental energia eòlica.....	64
6.2.2	Impacte ambiental energia Solar Fotovoltaica.....	67
7	LEGISLACIÓ ESPANYOLA RESPECTE LES ENERGIES RENOVABLES	69
8	CONCLUSIONS.....	70
9	BIBLIOGRAFIA.....	73

ÍNDIX DE FIGURES

Figura 1: Aerogenerador WINDPITCH.	13
Figura 2: Panell solar SR5-36.	13
Figura 3: Bateria 12 V RS 537-5444.	14
Figura 4: Esquema adaptadors.	15
Figura 5: Regulador SunGuard.	16
Figura 6: Adaptador tensió LM7805.	17
Figura 7: Divisors de tensions.	18
Figura 8: Esquema connexió telefònica.....	23
Figura 9: Arduino GSM Shield.....	24
Figura 10: GSM Modem Wavecom Q2303A.	25
Figura 11: Arduino UNO.....	26
Figura 12: Esquema Arduino UNO.....	28
Figura 13: Esquema maqueta pràctica.....	29
Figura 14: Maqueta pràctica a Rubió.	30
Figura 15: Logo LabWindows.....	32
Figura 16: Logo Arduino.....	32
Figura 17: Esquema de funcionament programa Arduino.....	33
Figura 18: Programa Arduino	34
Figura 19: Esquema de funcionament programa Labwindows	38
Figura 20: Esquema de funcionament detallat de la funció Rebre.....	39
Figura 21: Programa Labwindows.....	41
Figura 22: Esquema maqueta teòrica.....	47

Figura 23: Gràfic Tensió - Intensitat cel·la solar	50
Figura 24: ANEMO 4403.	52
Figura 25: Gràfic senyal sortida ANEMO 4403.	52
Figura 26: Cobertura 2g al territori Espanyol.	56
Figura 27: Inici Labwindows	60
Figura 28: Directori i fitxer de dades	60
Figura 29: Panell Labwindows.....	61
Figura 30: Dades fitxer	61

ÍNDIX DE TAULES

Taula 1: Característiques SR5-36 Solar Panel.....	13
Taula 2: Especificacions tècniques Regulador Sunguard 4 A 12 V Morningstar.....	16
Taula 3: Característiques tècniques Arduino UNO.....	27
Taula 4: Taula de dades.....	30
Taula 5: Modes baix consum Arduino.	48

1 INTRODUCCIÓ

L'objectiu d'aquest projecte és el disseny d'una maqueta capaç d'obtenir informació de dos sensors (eòlic i solar) per a permetre decidir la viabilitat de la instal·lació d'equipament energètic renovable en una ubicació concreta.

Aquesta maqueta comptarà amb el maquinari necessari com poden ser; sensors, circuit imprès, microcontrolador, elements de tractament de senyals, etc. I també disposarà del programari propi que requereixi la recollida d'informació proporcionada pels sensors. Aquesta part del projecte serà la que es portarà a la pràctica.

La segona part serà l'estudi teòric que pretén proporcionar total autonomia a la maqueta tant pel que fa a l'alimentació requerida com a la connexió sense fils per a la seva comunicació amb l'exterior, això també requerirà de modificacions al programa per optimitzar el consum de la maqueta.

Tot això estarà acompanyat d'un estudi de viabilitat i de condicions necessàries per poder implantar finalment un equipament energètic renovable. A més a més, un resum de les implicacions ambientals que aquestes energies suposen. I tot això, sense oblidar la legislació vigent referent a aquest tema, decisiva altre cop en termes de viabilitat econòmica.

2 ANTECEDENTS

Pel que fa a les energies renovables podem trobar multitud d'estudis referents a la viabilitat d'aquestes en funció de diversos factors relatius a l'energia en qüestió (en aquest projecte ens centrem en la eòlica i la solar), aquests factors poden ser:

En el cas de l'energia solar; la latitud, la zona climàtica, les hores d'exposició a radiació solar, l'angle d'incidència i l'orientació en són alguns dels més importants.

Pel que fa a l'energia eòlica; la intensitat del vent n'és la clau juntament amb les hores d'aplicació dintre de marges d'aquest, aquests factors són afavorits per les característiques de la seva ubicació. L'altitud i els obstacles de l'entorn són els factors més importants. Per tant, a major altitud i menor quantitat d'obstacles a les proximitats, major velocitat del vent. Zones pròximes al mar també en són molt favorables.

A part d'això, en termes de remuneració econòmica, la rendibilitat o taxa de retorn de la inversió és el principal varem a tenir en compte, aquest es calcula mitjançant el quocient entre el retorn net anual i la inversió inicial. El principal factor que juga a favor de la rendibilitat és el preu de venda de l'energia renovable, el qual és fluctuant i depèn molt de les subvencions en energies renovables vigents.

Pel que fa a aquest treball no es basarà en acotar la ubicació a grans trets, sinó que partint d'una ubicació escollida a partir d'altres mètodes més genèrics (per exemple, en el cas solar, buscant en taules la latitud i el número d'hores de radiació solar anuals), acotar aquesta ubicació a les més favorables i extreure'n les millors instal·lant la maqueta un cert període de temps en alguns punts de l'entorn per obtenir-ne dades pràctiques.



3 COS DEL PROJECTE

3.1 Components

Primerament es descriuran els components que formen part del treball per analitzar les seves característiques i veure quin valor aportaran al treball.

Es poden dividir en els següents apartats: Sensors, alimentació, adaptadors, connexió telefònica i arduino.

3.1.1 Sensors

Pel que fa als sensors, se'n farà servir dos, un per a mesurar l'energia solar i l'altre per l'eòlica.

Els sensors escollits són realment dispositius generadors d'energia elèctrica a partir d'altres tipus d'energies com poden ser en aquest cas la solar i l'eòlica. A la vegada serviran com a sensors ja que els dos proporcionen una tensió de sortida oscil·lant en funció la l'energia captada per aquests i a part també serà útil el fet de tractar-se de generadors ja que així es podrà aprofitar per alimentar el circuit necessari per a donar autonomia a la maqueta, tema del qual se'n parlarà a l'apartat 3.4 (estudi teòric per a la maqueta final).

3.1.1.1 Aerogenerador - WINDPITCH educational kit

Es tracta d'una turbina eòlica en miniatura [1] (veure Figura 1), dissenyada per avaluar la inclinació de les aspes i la forma d'aquestes per maximitzar la potència de sortida de la turbina en qüestió.

Aquest no serà l'objectiu requerit, però donat que la turbina proporciona una sortida en volts, servirà per acabar mesurant la tensió obtinguda a través de l'energia eòlica en forma de vent.

Dades importants:

Potència màxima de sortida = 1 w

Proporcionant un màxim de 12 V i 100 mA

És important mencionar que aquest sensor va ser escollit inicialment amb l'objectiu, a part de prendre dades, de proporcionar major similitud a un possible aerogenerador, dotant la maqueta de major utilitat de cara a futurs estudis amb aquesta, com podrien ser, el rendiment donat segons el tipus de pales o la inclinació d'aquestes. A posteriori (apartat 3.5) es podran trobar motius del perquè hagués sigut millor substituir aquest sensor per un anemòmetre.



Figura 1: Aerogenerador WINDPITCH.

3.1.1.2 Panell solar - SR5-36 Solar Panel

Es tracta d'un panell solar [2] (veure Figura 2) de dimensions reduïdes però que per actuar com a sensor són excessives. S'ha escollit aquest panell perquè pot ser reutilitzat per altres funcions degut a les seves característiques. Algunes de les més importants són:

Potència Nominal	5 W
Tensió d'alimentació Màx.	17 V
Longitud	194 mm
Alçada	28 mm
Longitud	194 mm
Intensitat a P. Màx.	300 mA
Tensió de circuit obert	21.6 V
Corrent de curtcircuit	330 mA
Pes Mín.	0.8 kg

Taula 1: Característiques SR5-36 Solar Panel.



Figura 2: Panell solar SR5-36.

3.1.2 Alimentació

Donat que l'energia eòlica i solar poden no ser constants i inclús interrompre's al llarg del dia ja sigui per núvols o poc vent, la maqueta requerirà d'una font d'alimentació constant que serà proporcionada a partir d'una bateria.

La idea principal és proporcionar energia a la bateria mitjançant el panell solar i a la vegada, aquesta donarà energia als components de la maqueta que requereixin d'alimentació.

3.1.2.1 Bateria - RS 537-5444

La bateria escollida és la: RS 537-5444 [3].

Una bateria d'àcid - plom de cicle intens (veure Figura 3), distribuïda per RS Pro. Presenta una capacitat de 1,2 Ah i una tensió nominal de 12 V. El seu pes és de 0,6 kg i les seves dimensions de 58 x 97 x 43 mm. Admet un rang de treball entre -20 i +60°C.



Figura 3: Bateria 12 V RS 537-5444.

3.1.3 Adaptadors

També es necessitaran un seguit d'adaptacions de tensió per a la càrrega de la bateria i per a la sortida d'aquesta cap a l'alimentació de la maqueta.

Primerament, s'utilitzarà un regulador comercial (veure 3.1.3.1) per realitzar la càrrega de la bateria i després de la bateria s'instal·larà un adaptador de tensió (veure 3.1.3.2) per ajustar-la a la necessitat de l'Arduino i el mòdem GSM (Sistema Global per a Comunicacions Mòbils).

A continuació es troba un petit esquema dels adaptadors i el seu connexionat (veure Figura 4):

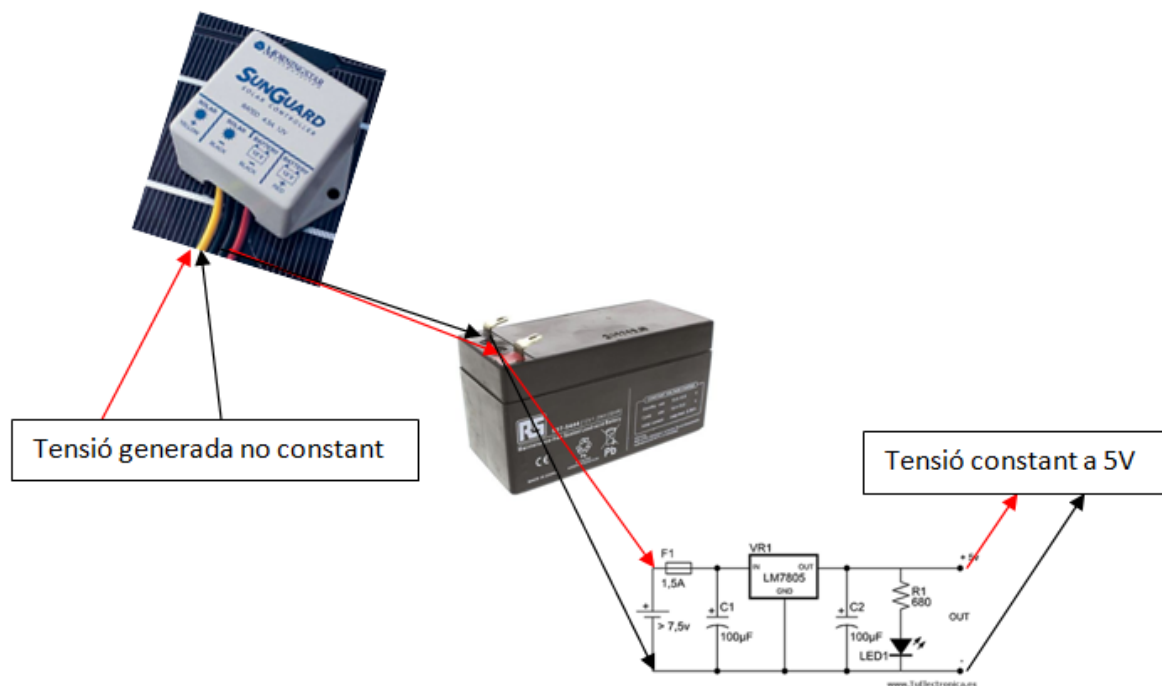


Figura 4: Esquema adaptadors.

3.1.3.1 Càrrega bateria - Regulador de càrrega SunGuard 4A 12V Morningstar

Es tracta d'un regulador comercial avançat [4] (veure Figura 5), petit i molt econòmic.



Figura 5: Regulador SunGuard.

Proporciona a la bateria una càrrega mitjançant font commutada (PWM). Aquest tipus de càrrega no aprofita tota la potència proporcionada per la bateria ja que quan aquesta es troba plena, el regulador talla certs cops per segons el contacte entre la placa i la bateria per evitar que aquesta es sobrecarregui. Aquest inconvenient es pot evitar instal·lant un regulador MPPT (Maximum power point tracking), però donat que l'objectiu no serà aprofitar la màxima potència, sinó simplement aprofitar aquesta energia produïda per alimentar algun component auxiliar de la maqueta, s'utilitzarà aquest regulador perquè és senzill, de pes reduït i econòmic.

El seu consum és aproximadament de 6 mA.

Específic per a exteriors, qualitat necessària per a la maqueta.

Entrada màxima	5'5 A (5 minuts)
Tensió del sistema	12 V
Tensió Solar Màx.	30 V
Tensió de regulació	14,1 V
Precisió	60 mV
Consum propi	6 mA
Compensació en temperatura	-28 mV/°C
Pèrdua de corrent en inversa	<10 µA
Temperatura d'operació	-40 a +85°C

Taula 2: Especificacions tècniques Regulador Sunguard 4 A 12 V Morningstar

3.1.3.2 Adaptador tensió bateria-circuit - LM7805

S'ha escollit un circuit [5] (veure Figura 6) que proporciona una tensió de sortida de 5 V estabilitzada, perfecte per alimentar l'arduino. La tensió d'entrada ha de ser continua i entre 7'5 i 35 V el qual també complim ja que estem alimentant a partir d'una bateria de 12V. L'excés de voltatge el dissipa el LM7805 en forma de calor, pel qual el fet de partir de 12V (un valor més proper a 7'5 que a 35 V) ajuda a que el LM7805 no es sobreescalfi excessivament. En aquest cas s'hauria d'afegir un petit dissipador de calor.

El circuit consta principalment del LM78XX un circuit integrat que regula la tensió al valor que trobem a XX, en el nostre cas 05 (5 V). També presenta dos condensadors de 100 μF que filtren la tensió, valor calculat a l'alça ja que el fabricant únicament recomana un mínim de 0'1 μF , la tensió d'aquests és recomanable que dupliqui la tensió de funcionament per tant 35 V seria un valor adequat i un fusible F1 de protecció en aquest cas de 1'5 A, ja que es treballarà amb valors inferiors a 1 A.

A part hi ha una resistència R1 i un LED a mode indicatiu de funcionament calculat a 2'5 V.

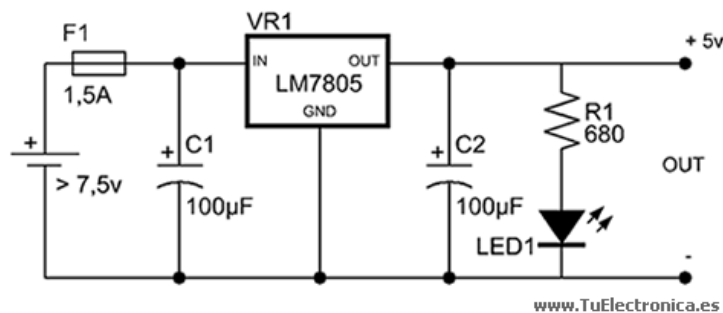


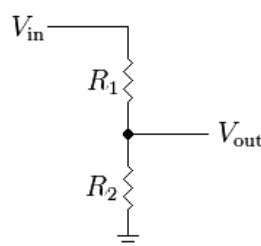
Figura 6: Adaptador tensió LM7805.

3.1.3.3 Divisor de tensió

La maqueta requereix una adaptació de la tensió proporcionada pels sensors de cara a ser llegida per la placa base a través de les entrades analògiques. Aquestes entrades accepten una tensió màxima de 5 V i per tant, haurem de reduir les tensions màximes proporcionades pels sensors, de 17 i 12 V, referents als sensors solar i eòlic respectivament.

Per tal d'escalar la tensió, utilitzarem dos divisors de tensió, un per a cada sensor. Com a tensió d'entrada prendrem els valors màxims teòrics esmentats anteriorment, i com a tensió de sortida prendrem 3 V, un valor inferior als 5 V de màxima acceptació dels sensors per cobrir-nos en cas de possibles sobretensions o de que les tensions proporcionades pels fabricants no siguin exactes.

A continuació podem observar l'esquema d'un divisor de tensió i la seva fórmula:



$$V_{Out} = \frac{R_2}{R_2 + R_1} * V_{In}$$

Fent servir l'equació i els valors abans citats es calculen les relacions entre resistències necessàries per als dos sensors (càlculs realitzats als següents fulls) i obtenim l'adaptador de tensions final (veure Figura 7) per a la nostra maqueta.

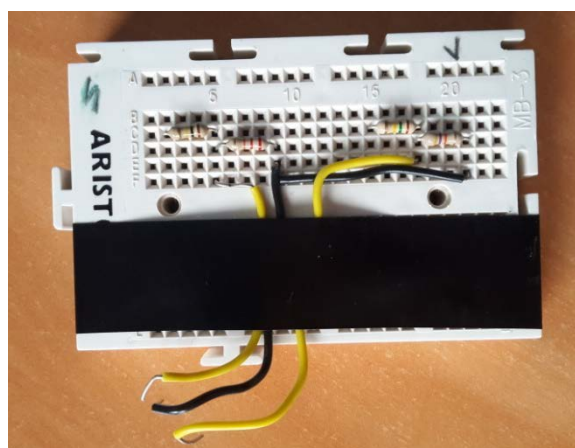


Figura 7: Divisors de tensions.

Càlcul divisor de tensió per al sensor solar:

$$V_{out} = \frac{R_1}{R_2 + R_1} * V_{in} \Rightarrow 3 = \frac{R_1}{R_2 + R_1} * 17$$

$$\frac{3}{17} * R_2 = R_1 - \left(\frac{3}{17} * R_1 \right)$$

$$\frac{3}{17} * R_2 = \frac{14}{17} * R_1$$

$$R_2 = \frac{14 * 17}{17 * 3} * R_1$$

$$R_2 = \frac{14}{3} * R_1 \Rightarrow R_2 = 4'6 * R_1$$

A part d'aquesta equació també s'ha de complir que els valors de les resistències siguin molt alts per aconseguir que l'energia consumida pel divisor sigui la menor possible, ja que a major resistència, menor serà la intensitat que circularà per aquestes i per tant la potència consumida també serà menor.

Prendrem valors de l'ordre de 10-100 KΩ.

Seguint aquestes dos premisses s'ha escollit una $R_1 = 22 \text{ K}\Omega$.

Aplicant aquesta R_1 a la fórmula;

$$R_2 = 4'6 * 22 \text{ K}\Omega = 102,6 \text{ K}\Omega$$

La resistència escollida és $R_2 = 100 \text{ K}\Omega$.

Resumint:

$$R_1 = 22 \text{ K}\Omega.$$

$$R_2 = 100 \text{ K}\Omega.$$

Per tant la tensió resultant després del divisor serà de:

$$V_{out} = \frac{R_1}{R_2 + R_1} * V_{in} \Rightarrow V_{out} = \frac{22 \text{ K}\Omega}{100 \text{ K}\Omega + 22 \text{ K}\Omega} * 17$$

$$V_{out} = 3,066 \text{ V}$$

Aquests càlculs s'han realitzat tenint en compte la tensió nominal provinent del panell solar, però si considerem que les resistències utilitzades tenen valors molt elevats, la càrrega total és molt baixa fet que es busca per desaprofitar el mínim d'energia possible, això ocasiona que la tensió que realment entra al divisor no serà la nominal sinó que tendirà més a ser propera a la de circuit obert, en aquest cas de 21'6 V.

Per tant cal comprovar que aquesta tensió, un cop passada pel divisor de tensió no superi els 5 V d'entrada màxima a les pots d'entrada analògica de l'arduino.

$$V_{out} = \frac{R_1}{R_2 + R_1} * V_{In} \Rightarrow V_{out} = \frac{22 \text{ K}\Omega}{100 \text{ K}\Omega + 22 \text{ K}\Omega} * 21'6$$

$$V_{out} = \mathbf{3'895 \text{ V}} < 5 \text{ V}$$

Càlcul divisor de tensió per al sensor eòlic:

$$V_{Out} = \frac{R_1}{R_2 + R_1} * V_{In} \Rightarrow 3 = \frac{R_1}{R_2 + R_1} * 12$$

$$\frac{3}{12} * R_2 = R_1 - \left(\frac{3}{12} * R_1 \right)$$

$$\frac{3}{12} * R_2 = \frac{9}{12} * R_1$$

$$R_2 = \frac{9 * 12}{12 * 3} * R_1$$

$$R_2 = \frac{9}{3} * R_1 \Rightarrow R_2 = 3 * R_1$$

A part d'aquesta equació i com s'ha comentat a l'apartat anterior es prendran valors de l'ordre de 10-100 KΩ.

Seguint aquestes dues premisses s'ha escollit una **$R_1 = 47 \text{ K}\Omega$** .

Aplicant aquesta R_1 a la fórmula;

$$R_2 = 3 * 47 \text{ K}\Omega = 141 \text{ K}\Omega$$

La resistència escollida és **$R_2 = 150 \text{ K}\Omega$** .

Resumint:

$$R_1 = 47 \text{ K}\Omega.$$

$$R_2 = 150 \text{ K}\Omega.$$

Per tant la tensió resultant després del divisor serà de:

$$V_{Out} = \frac{R_1}{R_2 + R_1} * V_{In} \Rightarrow V_{Out} = \frac{47 \text{ K}\Omega}{150 \text{ K}\Omega + 47 \text{ K}\Omega} * 12$$

$$V_{Out} = 2,863 \text{ V}$$

Tot i això és important tenir en compte que la resolució de l'Arduino és de 8 bits, per tant la tensió d'entrada (0 - 5 V) la dividirà en 2^8 valors, és a dir 256.

És important per no perdre tanta precisió, que el valor màxim d'entrada a l'Arduino s'ajusti al màxim a 5 V. Seguint aquesta premissa, recalculem les resistències i obtenim:

$$V_{Out} = \frac{R_1}{R_2 + R_1} * V_{In} \Rightarrow 5 = \frac{R_1}{R_2 + R_1} * 12$$

$$\frac{5}{12} * R_2 = R_1 - \left(\frac{5}{12} * R_1 \right)$$

$$\frac{5}{12} * R_2 = \frac{7}{12} * R_1$$

$$R_2 = \frac{7 * 12}{12 * 5} * R_1$$

$$R_2 = \frac{7}{5} * R_1 \Rightarrow R_2 = 1'4 * R_1$$

Seguint aquestes dues premisses s'ha escollit una $R_1 = 100 \text{ K}\Omega$.

Aplicant aquesta R_1 a la fórmula;

$$R_2 = 1'4 * 100 \text{ K}\Omega = 140 \text{ K}\Omega$$

La resistència escollida és $R_2 = 150 \text{ K}\Omega$.

Resumint:

$$R_1 = 100 \text{ K}\Omega.$$

$$R_2 = 150 \text{ K}\Omega.$$

Per tant la tensió resultant després del divisor serà de:

$$V_{Out} = \frac{R_1}{R_2 + R_1} * V_{In} \Rightarrow V_{Out} = \frac{100 \text{ K}\Omega}{150 \text{ K}\Omega + 100 \text{ K}\Omega} * 12$$

$$V_{Out} = 4,8 \text{ V}$$

3.1.4 Connexió telefònica

La maqueta requerirà d'una connexió amb l'exterior (veure Figura 8). Donat que la seva ubicació no serà constant i no té perquè ser de fàcil accés ni propera a punts d'accés telefònic, la connexió via cablejat queda descartada.

El tipus de comunicació que s'utilitzarà serà: GSM (Sistema Global per a Comunicacions Mòbils, originalment Groupe Spécial Mobile), és un estàndard desenvolupat per l'European Telecommunications Standards Institute (ETSI) per descriure els protocols de segona generació (2G) de xarxes digitals utilitzades pels telèfons mòbils.

El 2G va substituir el 1G (xarxes analògiques) i amb el temps s'ha millorat mitjançant el transport de paquets de dades a través de GPRS (General Packet Ràdio Services) i EDGE (Velocitats de Dades Millorades per l'Evolució de GSM o EGPRS). Posteriorment, el 3GPP (3rd generation partnership project) ha desenvolupat la tercera generació (3G) UMTS (Universal mobile telecommunications system) estàndards, seguits de la quarta generació (4G) LTE.

"GSM" és una marca comercial propietat de l'Associació GSM. També pot referir-se al codi més comú de veu en el passat, Full Rate.

Per realitzar la comunicació es necessitaran dos mòdems GSM, un per a la maqueta i l'altre per al PC. Explicats a continuació.

Les bandes o freqüències més comunes a Espanya són: GSM900, GSM1800 i 2100MHz.

La freqüència escollida és GSM 900MHz, banda utilitzada per les operadores mòbils espanyoles per oferir 3G degut a les seves bones qualitats de propagació.



Figura 8: Esquema connexió telefònica

3.1.4.1 Arduino GSM Shield

Arduino GSM Shield [6] (veure Figura 9) permet connectar un Arduino a Internet, utilitzant connexió GPRS.

Requereix únicament una targeta SIM amb cobertura GPRS.

Es tracta d'una plataforma de maquinari i programari lliure, per tant en tenim total accés.

El mòdem pot treballar a les següents freqüències: GSM850MHz, GSM900MHz, DCS1800MHz i PCS1900MHz.

La connexió amb l'Arduino UNO es pot realitzar a través dels pins 2 i 3 (programari serial) i pin 7 (reset).

L'alimentació de la placa és recomanable realitzar-la de manera externa, proporcionant-li entre 700 i 1000 mA



Figura 9: Arduino GSM Shield.

3.1.4.2 Mòdem GSM - GSM Modem Wavecom Q2303A Module USB Interface AT Commands SMS 900/1800MHz

Amb aquest mòdem [7] (veure Figura 10) es podrà connectar el PC a Internet mitjançant connexió GPRS i d'aquesta manera mantenir el PC en contacte amb l'arduino GSM Shield i per tant intercanviar dades.

Aquest producte ve equipat amb connexió USB, una antena DB de 80 mm, el cos del mòdem i un porta-SIM. Requereix lògicament d'una altra targeta SIM amb cobertura GPRS.

El mòdem pot treballar a les següents freqüències: GSM900MHz, GSM1800MHz.

Fabricant: Ostent

Dimensions: 6.5 x 5.4 x 1.3 "

Presenta tres modes de treball:

- 1- Asíncron, Transparent o No-Transparent, amb un baud rate de 14.400 bits/s
- 2- **Sèrie**, amb un baud rate de entre 300 i **115.200 bits/s**
- 3- Automàtic, amb un baud rate de entre 2.400 i 19.200 bits/s



Figura 10: GSM Modem Wavecom Q2303A.

3.1.5 Arduino

Finalment, i com a últim component, es troba l'arduino [6], on se centra el treball donat que recull el programa realitzat per dur a terme les funcions requerides.

Arduino és una plataforma electrònica de codi obert basada en maquinari i programari fàcil d'utilitzar. Està adreçat a qualsevol persona que pretengui realitzar projectes interactius. Utilitza el "Arduino programming language" basat en Wiring i el "Arduino Programari (IDE)" basat en Processing.

S'ha escollit arduino perquè es tracta d'una plataforma perfecta per realitzar prototips amb un cost relativament baix, amb programari lliure i multitud de llibreries que en faciliten el seu funcionament. A part també es tracta d'una plataforma molt flexible en el sentit de que permet programar a principiants i és molt interactiu. El fet de permetre extensions en el maquinari també n'és un punt a favor.

El model utilitzat per aquest treball és l'Arduino UNO (veure Figura 11), actualment només a la venda a USA tot i que aquí se'n pot aconseguir el seu germà, el Genuino UNO.



Figura 11: Arduino UNO.

3.1.5.1 Característiques tècniques

Característiques tècniques de l'Arduino UNO (veure Taula 3).

Microcontroller	ATmega328P
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limit)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Taula 3: Característiques tècniques Arduino UNO

3.1.5.2 Microcontrolador - ATmega328P

El Atmega328 AVR 8-bits és un circuit integrat d'alt rendiment que està basat en un microcontrolador RISC, combinant 32KB ISP flash una memòria amb capacitat de lectura mentre escriu, 1 KB de memòria EEPROM, 2KB SRAM memòria, 23 línies I/O de propòsit general, 32 registres del procés general, tres temporitzadors flexibles/comptadors de comparació, interrupcions internes i externes , programador de mode USART, una interfície serial orientada a byte de 2 cables, SPI de port serial, un convertidor A/D de 6 canals de 10 bits, un "watchdog timer" programable amb oscil·lador intern i cinc modes d'estalvi d'energia seleccionables per programari. El dispositiu opera entre 1.8 i 5.5 volts. A través de la implementació de poderoses instruccions en un únic cicle de rellotge, el dispositiu arriba a una resposta de 1 MIPS, balancejant de consum d'energia i la velocitat de processament.



3.1.5.3 Esquema Arduino UNO

Esquema de connexionat de l'Arduino UNO (veure Figura 12).

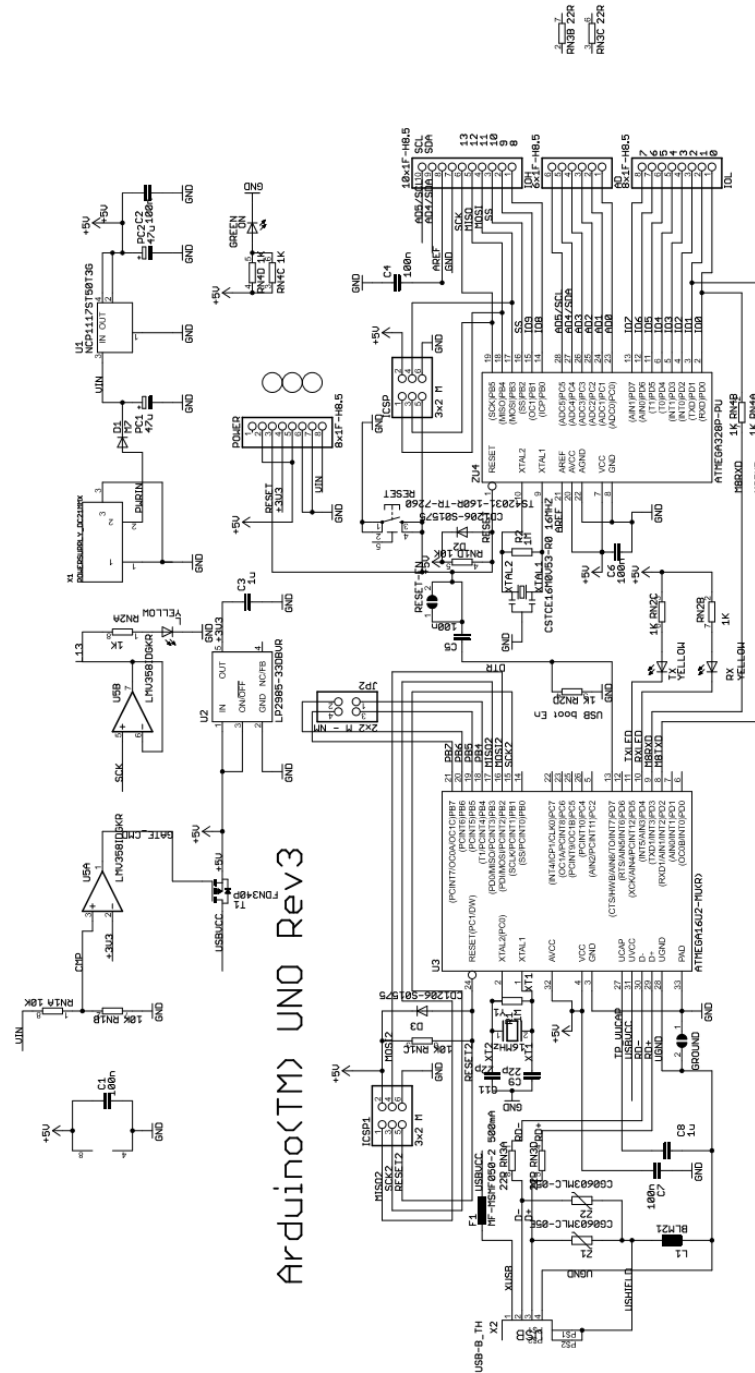


Figura 12: Esquema Arduino UNO

3.2 Estudi pràctic maqueta

En aquest apartat es descriu el circuit portat a la pràctica per comprovar el correcte funcionament de la maqueta al qual s'aplicarà el programa que es troba al següent apartat (3.3). Per finalitzar la maqueta i permetre el seu ús a peu de camp és necessari aplicar les millores descrites a l'apartat 3.4 (Estudi teòric per a la maqueta final).

El circuit portat a la pràctica està format per dos generadors, un d'eòlic i un de solar, que actuaran com a sensors per a prendre dades, un divisor de tensió el qual ens ajustarà la tensió proporcionada pels sensors als valors requerits per les entrades analògiques de l'arduino, que n'és el quart element. L'arduino tractarà els valors rebuts primerament amb el seu ADC (analog to digital converter) intern per convertir-los en digitals i poder-los processar, emmagatzemar i posteriorment enviar. Les entrades analògiques que s'utilitzen són la 0 i la 4 però es poden canviar per programa. I finalment un ordinador, el qual realitza les peticions i recepcions de dades per al seu posterior emmagatzematge i visualització.

A continuació es pot observar un petit esquema de la maqueta (veure Figura 12).

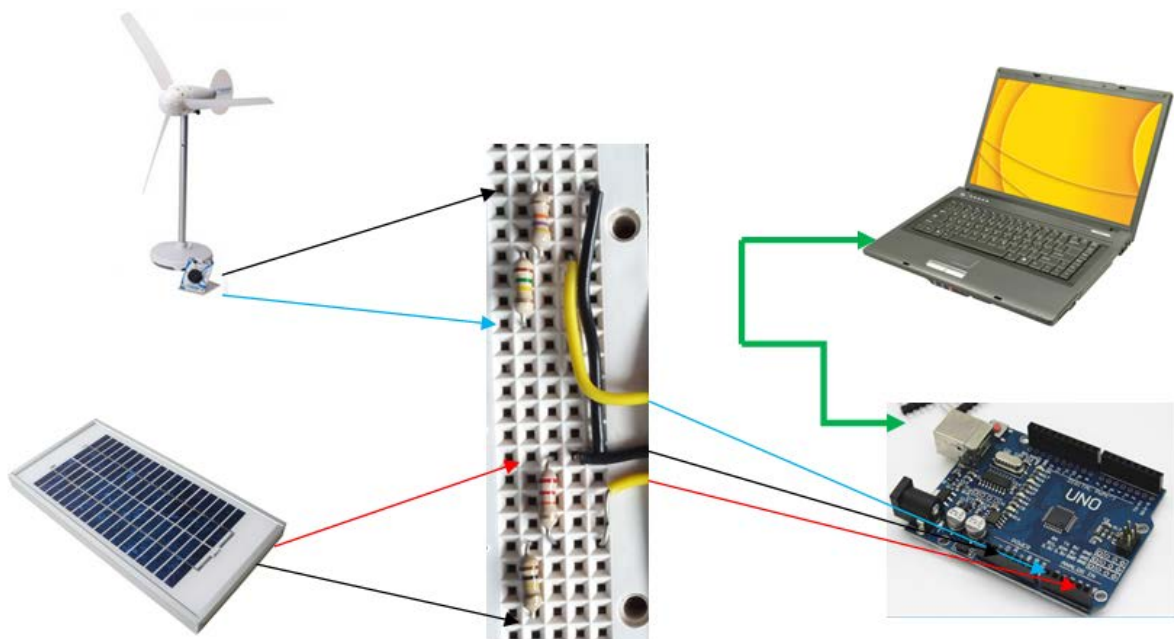


Figura 13: Esquema maqueta pràctica.



3.2.1 Dades Estudi Pràctic

El dia 04/06/2016 es van prendre dades utilitzant la maqueta pràctica a una ubicació favorable, en principi, per a instal·lar un parc eòlic-fotovoltaic: Rubió.

Per fer-ho es va utilitzar un programa de prova d'una durada de 30 minuts, el qual prenia dades cada 3,52 segons, aconseguint un total de 511 dades solars i 511 eòliques.

A continuació es pot observar la maqueta en funcionament (veure Figura 14), i un exemple de la taula de dades obtinguda (veure Taula 4).



Figura 14: Maqueta pràctica a Rubió.

	Temps	V Sol	V Vent			
Temps=	0	20,01	0,33			
Temps=	3,52	20,01	1,15			
Temps=	7,04	20,01	1,73			
Temps=	10,56	20,01	1,64			
Temps=	14,08	20,01	1,4		Mitjana SOL:	Mitjana VENT:
Temps=	17,6	20,01	1,4		19,693 V	1,626 V
Temps=	21,12	20,01	1,15			
Temps=	24,64	19,9	0,74			
Temps=	28,16	20,01	3,04			
Temps=	31,68	20,01	4,11			
Temps=	35,2	20,01	2,88		Total Dades/Sensor:	
Temps=	38,72	20,01	3,37		511	
Temps=	42,24	20,01	2,79			
Temps=	45,76	20,01	1,23			
Temps=	49,28	20,01	0,58			

Taula 4: Taula de dades.

3.3 Programa

El programa necessari es pot dividir en dues parts: la programació per al PC i la programació per a l'arduino.

En els següents apartats s'hi troben, prèvia elecció de programari, els programes utilitzats i una explicació de cadascun d'ells, però primerament es defineix a grans trets el funcionament del programa global.

L'objectiu del programa és recollir informació dels dos sensors cada cert temps i enviar-la al PC.

Per aconseguir-ho, l'arduino s'encarrega de recollir dades dels sensors cada 170 segons. Aquestes s'emmagatzemen a la EEPROM de manera que es va omplint la memòria amb una dada de cada sensor a cada posició.

Per altra banda, el PC cada 86.400 segons (1 cop al dia) envia una petició de dades a l'arduino, el qual al rebre tal comanda, envia pel port sèrie tota la informació de la qual disposa la EEPROM. Per fer-ho, es construeixen trames de dades i s'envien de forma asíncrona cap al PC una darrera de l'altra fins que s'ha enviat tota la informació disponible. Posteriorment, el PC rep les dades, les processa, les emmagatzema en un fitxer i les visualitza al panell d'usuari. Aquest procés es realitza repetitivament.

El motiu de fer servir la memòria EEPROM és que, en cas que es perdés l'alimentació de l'arduino, les dades no es perdrien.

L'elecció dels 86.400 segons és per enviar les dades al PC un cop al dia. Enviar les dades un cop al dia permet prendre dades dels sensors cada 170 segons, ja que la capacitat de la EEPROM de l'arduino UNO és de 1 Kb (1024 posicions). Tenint en compte que hem d'emmagatzemar dos valors cada vegada que prenem dades, es poden arribar a guardar dades 512 vegades. Amb els valors de temps escollits es prendran dades 508 vegades, el que significarà omplir 1016 posicions de memòria de les 1024 disponibles. Prendre dades dels sensors cada 170 segons és suficient, ja que no es creu necessari enviar-les més freqüentment, donat que les condicions meteorològiques influents sobre els sensors varien molt lentament (excloent excepcions puntuals).



3.3.1 Elecció Programari Arduino & Programari Ordinador

Per programar l'Arduino hi ha moltes opcions com poden ser: programació en assembler, llenguatge de baix nivell descartat per la seva dificultat; programació amb Keil, una de les millors eines per a la programació de microcontroladors, és molt potent i presenta multitud de llibreries però com a inconvenient trobem la seva complexitat i que és de pagament; i entre altres també hi ha el llenguatge propi d'arduino (veure Figura 16), un llenguatge d'alt nivell basat en Processing, molt similar a C++. Aquest, a diferència del Keil, és gratuït. El llenguatge de programació Arduino és l'escollit entre altres motius perquè és apte per a iniciar-se en la programació de prototips degut a la seva simplicitat i a la gran quantitat de llibreries que en faciliten la seva programació. Un inconvenient d'aquest programari podria ser que, com que no es programa en assembler, la utilització de llibreries produeix un retràs de l'ordre de microsegons en l'execució d'instruccions, significatiu per a segons quines aplicacions d'adquisició de dades amb temps molt reduïts però en el cas de la nostra maqueta aquest factor no és rellevant.

Per altra banda, per programar l'ordinador també hi ha multitud d'opcions, algunes d'elles treballades durant el grau com poden ser: Labwindows, Labview, Matlab, Python, etc. Dues de les més recents, utilitzades a informàtica industrial (Labwindows) i a instrumentació electrònica (Labview), aporten el programari necessari per a l'aplicació que se li pretén donar. Finalment l'elecció ha estat Labwindows (veure Figura 15) únicament per afinitat. Un punt fort de Labwindows en pot ser la interfície d'usuari, ja que proporciona un panell molt simple i funcional. Labwindows requereix de llicència de pagament, per aquest motiu i gràcies a la col·laboració de la UPC, s'ha demanat una llicència anual gratuïta com a alumne a National Instruments.



Figura 15: Logo LabWindows

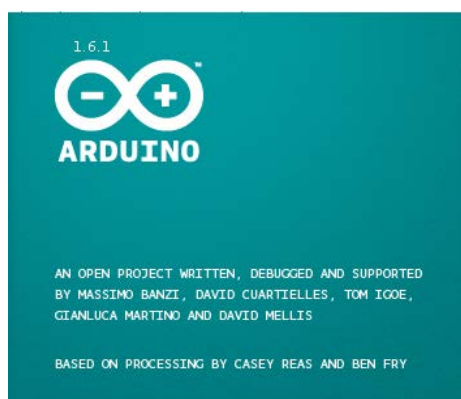


Figura 16: Logo Arduino



3.3.2 Programa Arduino

Esquema de funcionament (veure Figura 17):

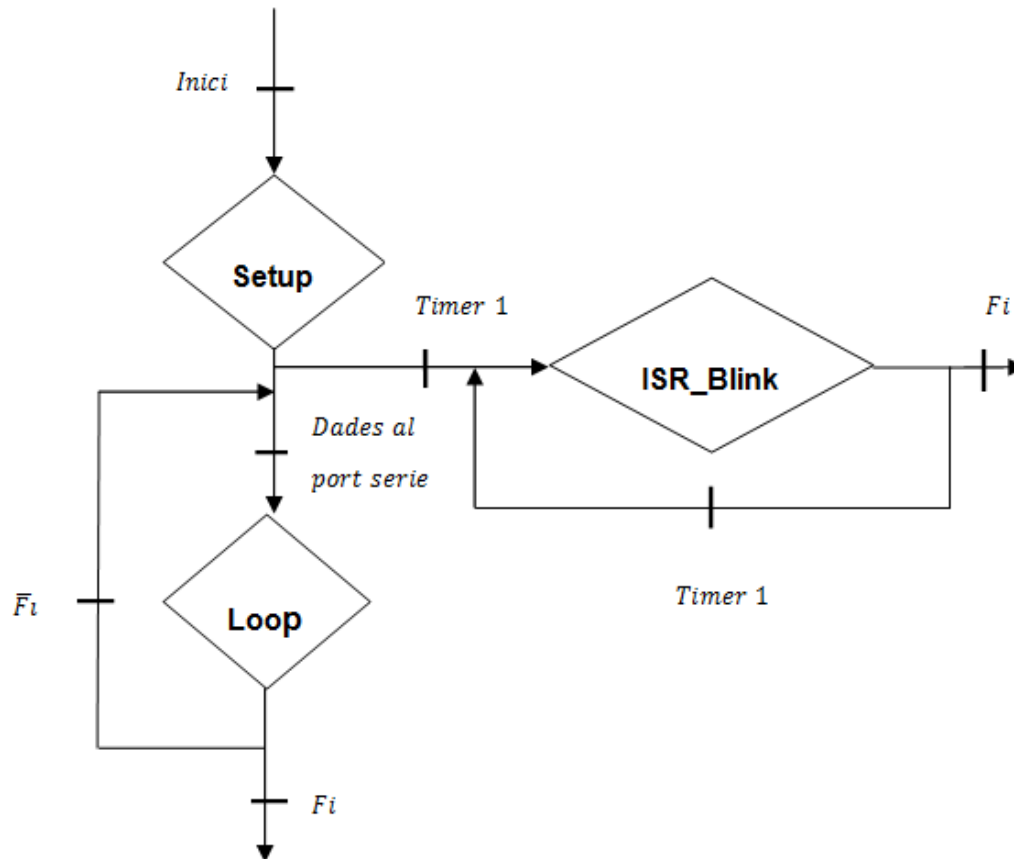


Figura 17: Esquema de funcionament programa Arduino.

On:

Setup; Programació de la interrupció i del port sèrie.

Loop; Transferència de dades Asíncrona amb el PC receptor mitjançant el port sèrie.

ISR_Blink; Rutina de servei d'interrupció que llegeix les dades dels sensors i les emmagatzema a la EEPROM de l'Arduino.

Codi del programa:

```
#include <EEPROM.h>
#include <TimerOne.h>
int addr=0;
int index = 0;
int estat = LOW;
int valor = 0;
int LED=13;

void setup() {
  addr = 0;
  index = 0;
  Timer1.initialize(170000000); // Dispara cada 170 segons
  Timer1.attachInterrupt(ISR_Blink); // Activa la interrupcion y la asocia a ISR_Blink
  Serial.begin(115200);
}

void loop() {
  int Cont;
  int posicio;
  int valS, valV;
  if (Serial.available())
  {
    Cont = Serial.read();
    switch (Cont)
    {
      case 49:
        noInterrupts();
        Serial.println(index);
        posicio = 0;
        while ( posicio < addr)
        {
          while (Serial.available());
          Cont = Serial.read();
          valS = EEPROM.read(posicio);
          posicio = posicio + 1;
          valV = EEPROM.read(posicio);
          posicio = posicio + 1;
          Serial.print('I');
          Serial.print(valS); // print as an ASCII-encoded decimal
          Serial.print(' ');
          Serial.println(valV); // print as an ASCII-encoded decimal
        }
        index = 0;
        addr = 0;
        interrupts();
        break;
      default:
        break;
    }
  }
}

void ISR_Blink(){
  long analogValueS, analogValueV;
  int valS, valV;
  if (estat == LOW)
    estat = HIGH;
  else
    estat = LOW;
  digitalWrite(LED, estat);
  analogValueS = analogRead(0);
  analogValueV = analogRead(4);
  valS = int(analogValueS >> 2);
  valV = int(analogValueV >> 2);
  EEPROM.write(addr, valS); // value: the value to write, from 0 to 255 (byte)
  addr++;
  if (addr == EEPROM.length()) //1024 posicions de memoria
  {
    addr = 0;
  }
  EEPROM.write(addr, valV); // value: the value to write, from 0 to 255 (byte)
  addr++;
  if (addr == EEPROM.length()) //1024 posicions de memoria
  {
    addr = 0;
  }
  index = index + 1;
}
```

Figura 18: Programa Arduino

A continuació s'explica el programa pas a pas:

```
#include <EEPROM.h>
#include <TimerOne.h>
```

Primerament s'inclouen les llibreries necessàries.

```
int addr=0;
int index = 0;
int estat = LOW;
int LED=13;
```

A continuació es declaren les variables globals.

```
void setup()
{
    addr = 0;
    index = 0;
    Timer1.initialize(170000000); // Dispara cada 170 segons
    Timer1.attachInterrupt(ISR_Blink); // Activa la interrupcion y la asocia a ISR_Blink
    Serial.begin(115200);
}
```

El *setup* s'encarrega de programar la interrupció *ISR_Blink* (interrupció interna), que dispararà cada 170 segons. També es programa el port sèrie a un baud rate de 115.200 (el màxim permès per l'Arduino). A part es reinicien a 0 les variables *addr* i *index* cada cop que carreguem el programa.

```
void loop()
{
    int Cont;
    int posicio;
    int valS, valV;
    if (Serial.available())
    {
        Cont = Serial.read();
    }
}
```

El *loop()* serà la funció que permetrà transmetre la informació a través del port sèrie. Primerament es declaren algunes variables internes i després es mira si hi ha dades al port sèrie, en cas afirmatiu, es llegeixen i s'emmagatzemen a la variable *Cont*, la qual permet fer el programa asíncron.

```

switch (Cont)
{
  case 49:
    noInterrupts();
    Serial.println(index);
    posicio = 0;

```

En cas que la variable *Cont* sigui igual a 49, es deshabiliten les interrupcions, s'envia la variable *index* (encarregada de definir el número de cadenes que s'enviaran) pel port sèrie i es reinicia la variable *posicio* a 0.

```

while ( posicio < addr)
{
  while (Serial.available());
  Cont = Serial.read();
  valS = EEPROM.read(posicio);
  posicio = posicio + 1;
  valV = EEPROM.read(posicio);
  posicio = posicio + 1;
  Serial.print('I');
  Serial.print(valS); // print as an ASCII-encoded decimal
  Serial.print(' ');
  Serial.println(valV);
}

```

Mentre *posicio* sigui més petit que *addr* i sempre i quant hi hagi dada al port sèrie (sinó esperem fins que arribi), es llegeix la variable *Cont* (per treure-la de la cua) la qual permet la transmissió asíncrona. Seguidament es llegeixen les dades emmagatzemades a la EEPROM i s'incrementa la variable *posicio* per a cada dada llegida. Posteriorment s'envia pel port sèrie una cadena formada per 'I' + 'valS' + ' ' + 'valV' + '\n', aquest últim és un salt de pàgina en C+.

```

        index = 0;
        addr = 0;
        interrupts();
        break;
    default:
        break;
  }
}
}

```

Un cop s'han enviat totes les dades que hi havia a la EEPROM, es posen a 0 les variables *index* i *addr* i s'habiliten les interrupcions.

```
void ISR_Blink()
{
    long analogValueS, analogValueV;
    int valS, valV;
    if (estat == LOW)
        estat = HIGH;
    else
        estat = LOW;
    digitalWrite(LED, estat);
}
```

La funció *ISB_Blink()* s'encarrega de llegir els valors que proporcionen els sensors i emmagatzemar-los a la EEPROM. Això es realitzarà cada cop que salti la interrupció, definida al *setup*, en aquest cas, cada 170 segons. El motiu d'emmagatzemar-los a la EEPROM és que en cas de que es perdés l'alimentació de l'Arduino per qualsevol motiu, les dades no es perdrien.

Es defineixen algunes variables locals i a continuació trobem un canvi en el *LED 13*, led que presenta l'Arduino, simplement a mode de comprovació de que la interrupció funciona correctament.

```
analogValueS = analogRead(0);
analogValueV = analogRead(4);
valS = int(analogValueS >> 2);
valV = int(analogValueV >> 2);
```

Seguidament, es llegeixen els valors de les entrades analògiques 0 i 4 provinents dels sensors, entre 0 i 5 V, però com el ADC és de 10 bits i la EEPROM només permet valors de 8 bits, es converteixen aquests valors a 8 bits.

```
EEPROM.write(addr, valS); // value: the value to write, from 0 to 255 (byte)
addr++;
if (addr == EEPROM.length()) // 1024 posicions de memoria
{
    addr = 0;
}
EEPROM.write(addr, valV); // value: the value to write, from 0 to 255 (byte)
addr++;
if (addr == EEPROM.length()) // 1024 posicions de memoria
{
    addr = 0;
}
index = index + 1;
}
```

Finalment, s'escriuen els valors llegits a la EEPROM, a la posició *addr* i s'incrementa la variable *addr* per continuar escrivint els següents valors a les següents posicions. Per últim, s'incrementa la variable *index*, la qual serveix per seguir el número de línies que s'envien al PC mitjançant la funció anterior "*loop()*".

3.3.3 Programa Labwindows

Esquema de funcionament (veure Figura 19):

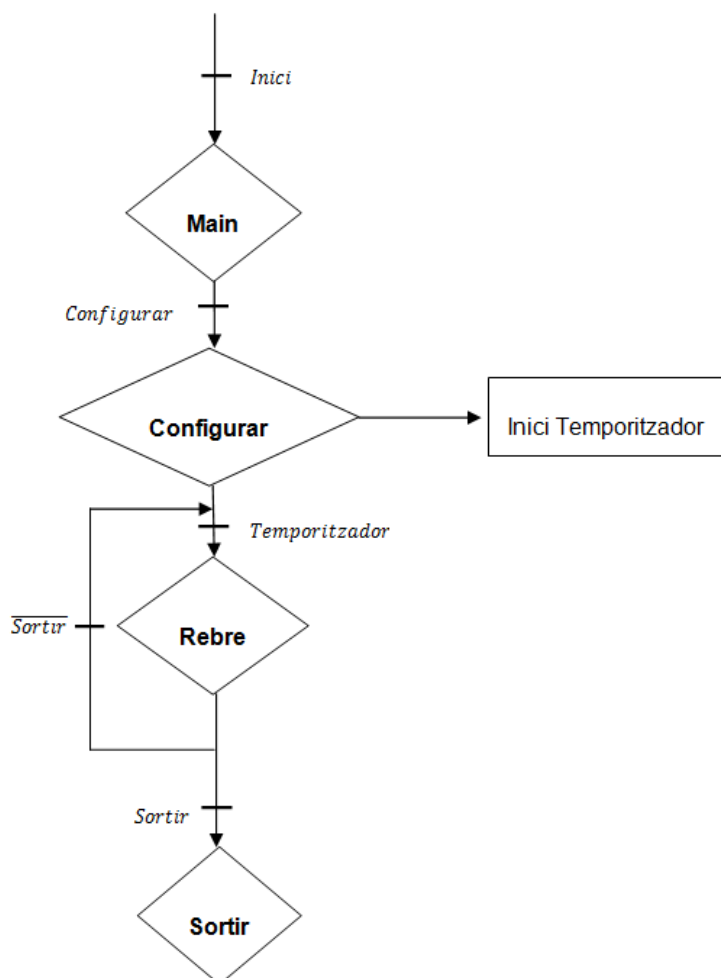


Figura 19: Esquema de funcionament programa Labwindows

On:

Main; Carrega el panell, demana nom i ubicació per al fitxer de text i inicia la interfície d'usuari.

Configurar; Llegeix el número de port que s'utilitzarà del panell d'usuari i configura el port sèrie. També inicia el temporitzador.

Rebre; Funció encarregada de la comunicació amb l'Arduino a través del port sèrie. També emmagatzema les dades rebudes en un fitxer i les dibuixa al panell d'usuari.

Sortir; Tanca el port i finalitza el programa.

Esquema de funcionament detallat de la funció Rebre (veure Figura 20):

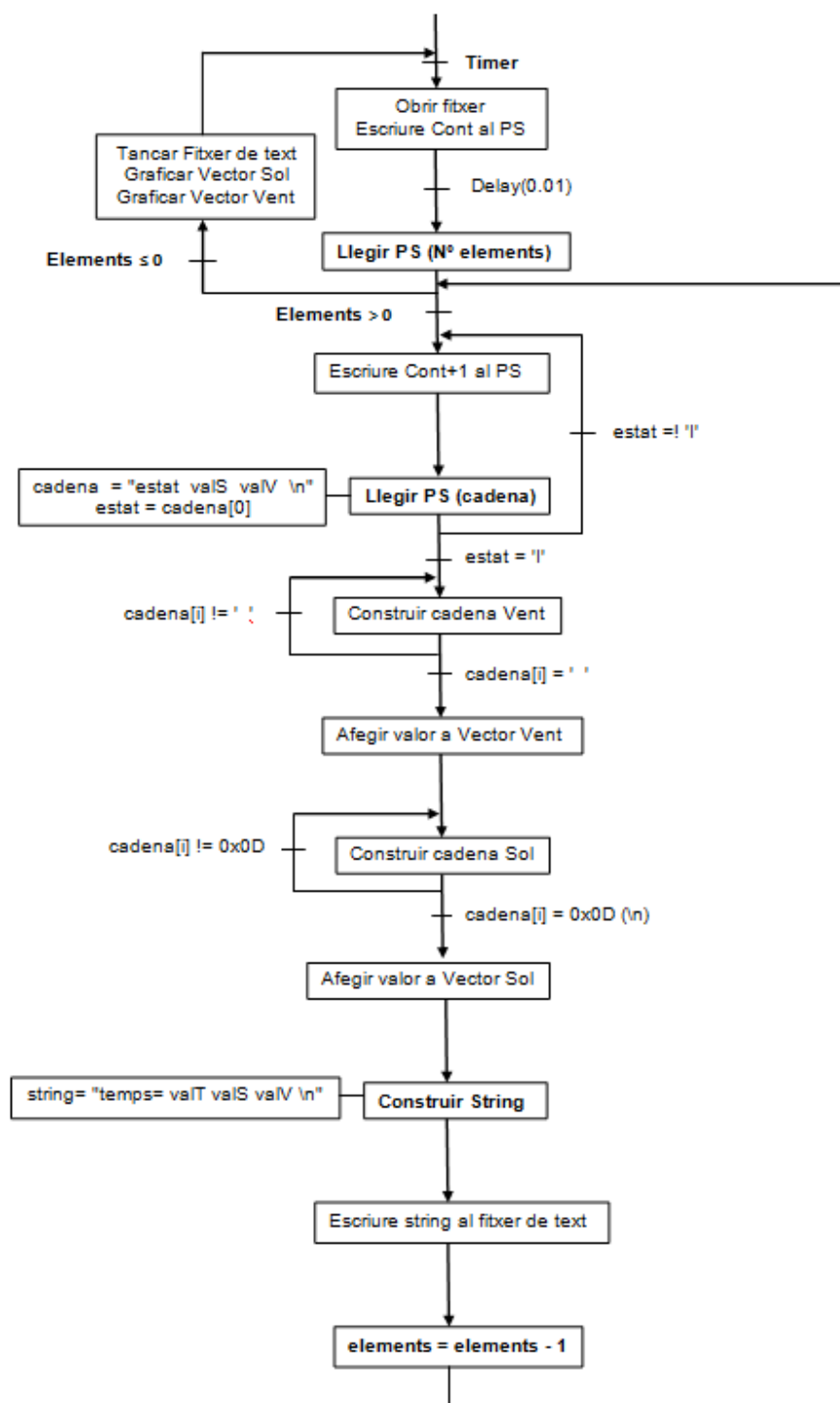


Figura 20: Esquema de funcionament detallat de la funció Rebre.

Codi del programa:

```
#include <rs232.h>
#include <formatio.h>
#include <utility.h>
#include <ansi_c.h>
#include <cvirte.h>
#include <userint.h>
#include "p2.h"

static int pantalla;

int port;
char TCCR2B;
int FitxerHandle;
char NomFitxer[260];
float temps=0;

int main (int argc, char *argv[])
{
    if (InitCVRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((pantalla = LoadPanel (0, "p2.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (pantalla);
    FileSelectPopup ("", "*.*", "", "", VAL_SAVE_BUTTON, 0, 0, 1, 0, NomFitxer);
    RunUserInterface ();
    DiscardPanel (pantalla);
    return 0;
}

int CVICALLBACK Sortir (int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            CloseCom (port);
            QuitUserInterface (0);
            break;
    }
    return 0;
}

int CVICALLBACK Configurar (int panel, int control, int event,
                            void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal (pantalla, PANEL_PORT, &port);
            OpenComConfig (port, "", 115200, 0, 8, 1, 512, 512);
            SetCtrlAttribute (pantalla, PANEL_TIMER, ATTR_ENABLED, 1);
            break;
    }
    return 0;
}
```



```
int CVICALLBACK Rebre (int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
{
    char cadena[32];
    char cad[32];
    char bufVent[10], bufSol[10], bufTemps[10];
    char cont= 49;
    int nCar = 0;
    int valorI = 0;
    double VectorVENT[10000]={0};
    double VectorSOL[10000]={0};
    int index = 0;
    unsigned int elements;
    int i, j;
    char estat;
    double valorF=0;
    int contador;
    char string[50];
    int nCadena;

    switch (event)
    {
        case EVENT_TIMER_TICK:
            FitzerHandle = OpenFile (NomFitzer, VAL_WRITE_ONLY, VAL_APPEND, VAL_ASCII);
            ComWrtByte(port, cont);
            Delay(0.01);
            nCar = ComRdTerm(port, cadena, 32, 0x0a);
            Scan(cadena, "%s%i", &elements);
            contador=0;
            while (elements>0)
            {
                ComWrtByte(port, cont+1);
                nCar = ComRdTerm (port, cadena, 32, 0x0a);
                estat = cadena[0];
                if (estat == 'I')
                {
                    // Lectura del Sol
                    i = 1;
                    j = 0;
                    while (cadena[i] != ' ')
                    {
                        cad[j] = cadena[i];
                        i++; j++;
                    }
                    cad[j] = 0;
                    Scan(cad, "%s%i", &valorI);
                    valorF=(((double)valorI/(255.0*(3.065/5.0)))*17.0);
                    VectorSOL[index] = valorF;
                    Fmt (bufSol, "%s<%f[p2]", valorF);

                    // Lectura del Vent
                    j = 0;
                    while (cadena[i] != 0x0D)
                    {
                        cad[j] = cadena[i];
                        i++; j++;
                    }
                    cad[j] = 0;
                    Scan(cad, "%s%i", &valorI);
                    valorF=(((double)valorI/(255.0*(2.863/5.0)))*12.0);
                    VectorVENT[index] = valorF;
                    Fmt (bufVent, "%s<%f[p2]", valorF);
                    index++;
                    // Construccio string
                    strcpy(string,"Temps= ");
                    Fmt (bufTemps, "%s<%f[p2]", temps);
                    temps= temps + 170 ;
                    strcat(string, bufTemps);
                    strcat(string, " ");
                    strcat(string, bufSol);
                    strcat(string, " ");
                    strcat(string, bufVent);
                    strcat(string, "\n");
                    nCadena = strlen(string);
                    WriteFile (FitzerHandle, string, nCadena);
                    elements--;
                }
            }

            CloseFile ( FitzerHandle );

            PlotStripChart (pantalla, PANEL_STRIPCHART_SOL, VectorSOL, index, 0, 0, VAL_DOUBLE);
            PlotStripChart (pantalla, PANEL_STRIPCHART_VENT, VectorVENT, index, 0, 0, VAL_DOUBLE);
            break;
    }

    return 0;
}
```

Figura 21: Programa Labwindows

A continuació s'explica el programa pas a pas:

```
#include <rs232.h>
#include <formatio.h>
#include <utility.h>
#include <ansi_c.h>
#include <cvirte.h>
#include <userint.h>
#include "p2.h"
```

Primerament s'inclouen les llibreries necessàries.

```
static int pantalla;
int port;
char TCCR2B;
int FitxerHandle;
char NomFitxer[260];
float temps=0;
```

A continuació es declaren les variables globals.

```
int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((pantalla = LoadPanel (0, "p2.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (pantalla);
    FileSelectPopup ("", "**.*", "", "", VAL_SAVE_BUTTON, 0, 0, 1, 0, NomFitxer);
    RunUserInterface ();
    DiscardPanel (pantalla); return 0;
}
```

La funció *main()* s'encarrega de carregar el panell, demana nom i ubicació per al fitxer de text i inicia la interfície d'usuari.

```
int CVICALLBACK Sortir (int panel, int control, int event, void *callbackData, int eventData1,
                        int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            CloseCom (port);
            QuitUserInterface (0);
            break;
    }
    return 0;
}
```

La funció *Sortir()* tanca el port i finalitza el programa.

```
int CVICALLBACK Configurar (int panel, int control, int event, void *callbackData,
                           int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal (pantalla, PANEL_PORT, &port);
            OpenComConfig (port, "", 115200, 0, 8, 1, 512, 512);
            SetCtrlAttribute (pantalla, PANEL_TIMER, ATTR_ENABLED, 1);

            break;
    }
    return 0;
}
```

La funció *Configurar()*, llegeix el número de port que s'utilitzarà del panell d'usuari i configura el port sèrie a un baud rate de 115200. També inicia el temporitzador que reportarà a la funció *Rebre()*, en aquest cas cada 86400 segons.

```
int CVICALLBACK Rebre (int panel, int control, int event, void *callbackData,
                      int eventData1, int eventData2)
{
    char cadena[32];
    char cad[32];
    char bufVent[10], bufSol[10], bufTemps[10];
    char cont= 49;
    int nCar = 0;
    int valorI = 0;
    double VectorVENT[10000]={0};
    double VectorSOL[10000]={0};
    int index = 0;
    unsigned int elements;
    int i, j;
    char estat;
    double valorF=0;
    int contador;
    char string[50];
    int nCadena;
```

La funció *Rebre()* és l'encarregada de la comunicació amb l'Arduino. Aquí es centra el cos del programa. Primerament, es declaren les variables locals.

```
switch (event)
{
    case EVENT_TIMER_TICK:
        FitxerHandle=OpenFile(NomFitxer, VAL_WRITE_ONLY,
                              VAL_APPEND, VAL_ASCII);

        ComWrtByte(port, cont);
        Delay(0.01);
        nCar = ComRdTerm(port, cadena, 32, 0x0a);
        Scan(cadena, "%s>%i", &elements);
        contador=0;
```

Cada cop que el temporitzador configurat a la funció *Configurar()* arriba als 86400 segons programats, es produeix el que labwindows anomena *EVENT_TIMER_TICK*, que fa entrar dins del case. Seguidament, s'obre el fitxer de text on s'emmagatzemaran les dades, s'escriu la variable *cont* (49) al port sèrie i es llegeix la *cadena* que arriba al port sèrie des de l'Arduino. Finalment, es transforma aquesta en un enter anomenat *elements* i es reinicia la variable *contador* a 0.

```

while (elements>0)
{
    ComWrtByte(port, cont+1);
    nCar = ComRdTerm (port, cadena, 32, 0x0a);
    estat = cadena[0];
    if (estat == 'I')
    {
        // Lectura del Sol
        i = 1;
        j = 0;
        while (cadena[i] != ' ')
        {
            cad[j] = cadena[i];
            i++;
            j++;
        }
    }
}

```

En cas que la variable *elements* sigui major a 0 (això vol dir que hi ha x línies de text a rebre pel port sèrie), s'envia *cont+1* al port sèrie (això es fa per fer la comunicació asíncrona, per tal de no perdre cap dada). Seguidament es llegeix *cadena* del port sèrie i s'emmagatzema la primera posició d'aquesta a la variable *estat*. Si *estat* és igual a "I", es realitza la lectura de la variable Vent que es troba emmagatzemada a la variable *cadena* des de la posició 1 fins que es troba un espai " ".

```

cad[j] = 0;
Scan(cad, "%s>%i", &valorI);
valorF=(((double)valorI/(255.0*(3.065/5.0)))*17.0);
VectorSOL[index] = valorF;
Fmt (bufSol, "%s<%.2f", valorF);

```

A continuació es posa un 0 al final de la variable *cad*, on hi ha el valor del Sol, per tal de limitar la cadena. Aquesta variable es transforma a un double per poder-la dibuixar a posteriori, s'afegeix al VectorSOL i es transforma en un string amb dos decimals per guardar-la al fitxer de text.

```

// Lectura del Vent
j = 0;
while (cadena[i] != 0x0D)
{
    cad[j] = cadena[i];
    i++;
    j++;
}
cad[j] = 0;
Scan(cad, "%s>%i", &valorI);
valorF=(((double)valorI/(255.0*(2.863/5.0)))*12.0);
VectorVENT[index] = valorF;
Fmt (bufVent, "%s<%.2f", valorF);
index++;

```

Es repeteix el codi anterior però per llegir la variable Vent que es troba entre l'espai " " i el final de trama "0x0D".

Finalment, s'incrementa la variable *index* la qual ordena els valors al VectorSOL i VectorVENT.

```

// Construcció string
strcpy(string, "Temps= ");
Fmt (bufTemps, "%s<%f[p2]", temps);
temps= temps + 170 ;
strcat(string, bufTemps);
strcat(string, " ");
strcat(string, bufSol);
strcat(string, " ");
strcat(string, bufVent);
strcat(string, "\n");
nCadena = strlen(string);
WriteFile (FitxerHandle, string, nCadena);
elements--;
    }
    CloseFile ( FitxerHandle );
    PlotStripChart (pantalla, PANEL_STRIPCHART_SOL, VectorSOL,
        index, 0, 0, VAL_DOUBLE);
    PlotStripChart (pantalla, PANEL_STRIPCHART_VENT, VectorVENT,
        index, 0, 0, VAL_DOUBLE) ;
    break;
}
return 0;
}

```

Un cop llegits els valors, es construeix la sentència que s'emmagatzemarà al fitxer de text, on s'hi afegeix el temps en que s'ha realitzat la lectura i els dos valors llegits en aquell instant. Seguidament s'escriu aquesta sentència al fitxer i es disminueix la variable *elements*.

Aquest procés de lectura de dades es repetirà mentre la variable *elements* sigui més gran que 0 (és a dir, mentre quedin valors per llegir a la EEPROM de l'Arduino).

Un cop s'han llegit tots els valors, es tanca el fitxer de text i es dibuixen els dos vectors on s'han carregat les dades; el del Sol i el del Vent.

3.4 Estudi teòric per a la maqueta final

En aquest apartat es realitza una ampliació del circuit portat a la pràctica explicat a l'apartat 3.1.5.1, amb l'objectiu de donar viabilitat a una futura posada en marxa in situ de la maqueta en les ubicacions escollides. Per fer possible aquest objectiu és necessari dotar la maqueta d'autonomia, tant pel que fa a l'alimentació com a la comunicació.

La comunicació de la maqueta amb l'exterior, en aquest cas amb el PC receptor d'informació no sempre es podrà fer mitjançant cablejat ja que la maqueta s'ha de poder situar en ubicacions remotes, per tant cal dotar la maqueta de comunicació sense fils. Els elements escollits per portar a terme aquesta funció són dos mòdems GSM, un emissor (Arduino GSM Shield) i un receptor (Mòdem GSM - GSM Modem Wavecom Q2303A Module USB Interface AT Commands SMS 900/1800MHz) que treballaran a una freqüència de 900 MHz.

L'alimentació es farà a través de l'energia proporcionada pel panell solar, el qual a part de realitzar la funció de sensor, també treballarà com a generador. S'acumularà aquesta energia produïda en una bateria (Bateria - RS 537-5444) mitjançant un regulador (Càrrega bateria - Regulador de càrrega SunGuard 4A 12V Morningstar) per ajustar la càrrega i no fer malbé la bateria. Finalment, a la sortida de la bateria s'hi instal·larà un circuit adaptador (Adaptador tensió bateria-circuit - LM7805) per proporcionar una tensió final estabilitzada a 5V. Aquesta tensió final serà la que permeti alimentar l'arduino (Arduino) i el mòdem emissor (Arduino GSM Shield).

A continuació es pot observar un petit esquema de com quedaria la maqueta final (Figura 22).

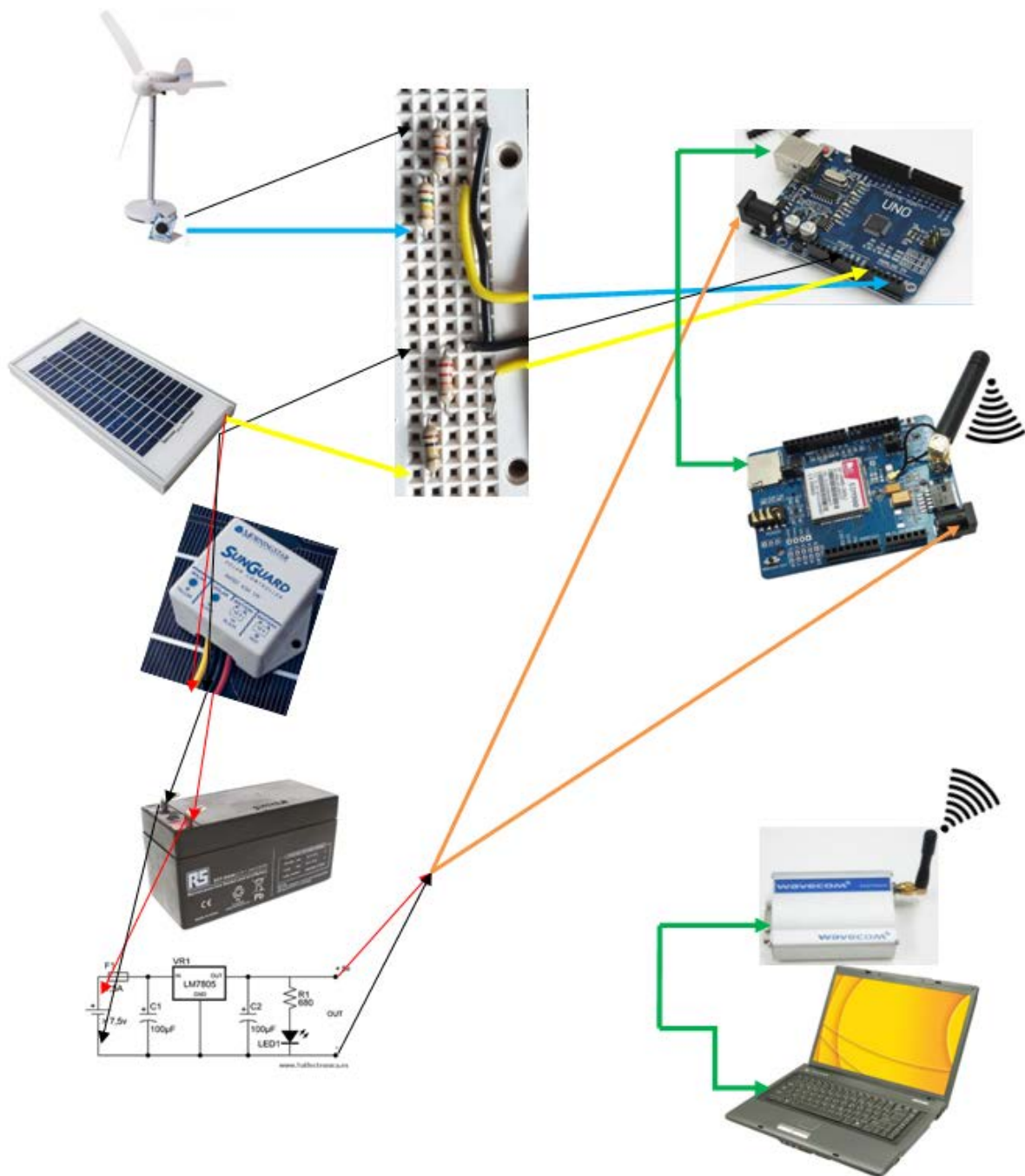


Figura 22: Esquema maqueta teòrica.

3.4.1 Modificacions programa final

Per tal d'optimitzar el consum de l'arduino, seria necessari utilitzar algun mode de baix consum. Els 5 modes de baix consum de l'Arduino són els següents [15]:

1. SLEEP_MODE_IDLE -> consum=15mA - el que menys estalvia
2. SLEEP_MODE_ADC -> consum=6.5mA
3. SLEEP_MODE_PWR_SAVE -> consum=1.62mA
4. SLEEP_MODE_STANDBY -> consum=0.84mA
5. SLEEP_MODE_PWR_DOWN -> consum=0.36mA - el que més estalvia

La següent taula (veure Taula 5) mostra les funcions que mantenen actives els modes de baix consum del arduino.

	clk CPU	clk Flash	clk Flash	clk ADC	clk ASY	clk principal	time r1	int0 int1	T W I	time r 2	SPM/ EEP ROM	ADC	WDT	Otro s
IDLE			X	X	X	X	X	X	X	X	X	X	X	X
ADC POWERED					X	X	X	X	X	X	X	X	X	
PWR_DOW N								X	X				X	
PWR_SAVE					X		X	X	X	X			X	
StandBy						X		X	X				X	

Taula 5: Modes baix consum Arduino.



La llibreria utilitzada per AVR sleep és la següent: **#include <avr/sleep.h>**

Les funcions que ens proporciona la llibreria AVR sleep són:

- **set_sleep_mode(mode):** aquí s'estableix el mode de baix consum escollit.
- **sleep_enable():** habilita el sistema per iniciar el mode de baix consum, però encara no està actiu.
- **sleep_mode():** activa el mode de baix consum.
- **sleep_disable():** desactiva el mode de baix consum.

El microcontrolador hauria de quedar "adormit", en baix consum, fins que rebés una interrupció, ja sigui interna per llegir i emmagatzemar dades, o externa, per enviar les dades emmagatzemades cap al PC. La interrupció interna ja la presenta el programa a partir del timer 1 per tant els modes SLEEP_MODE_STANDBY i SLEEP_MODE_PWR_DOWN queden descartades ja que inhabiliten aquest timer. La interrupció externa no apareix al programa, aquesta hauria de provenir de la GSM Shield pel port sèrie.

Entre els tres modes de baix consum que serviren, l'elegit seria el SLEEP_MODE_PWR_SAVE degut a que és el que menys consumeix.

Per tal de despertar el microcontrolador a partir de l'arribada de dades al port sèrie, es pot connectar el Pin 2 amb el Rx. Així a l'arribar dades al serial, baixaran els 5V pel Pin 2 i això dispararà la interrupció (INT 0), la qual s'utilitzaria per despertar l'Arduino.

Un cop despertat l'Arduino (fora del mode de baix consum) s'hauran de deshabilitar les interrupcions, ja que sinó cada dada que arribés pel serial produiria una nova interrupció i això faria fallar al programa. Un cop enviades totes les dades, es tornarien a habilitar les interrupcions i es reintroduiria el mode de baix consum fins a rebre una nova interrupció.

3.5 Viabilitat

Pel que fa a la viabilitat d'utilització d'aquesta maqueta s'han de tenir presents diferents factors. Tenint en compte que la maqueta utilitzada per realitzar estudis sobre la viabilitat de les energies renovables en ubicacions concretes seria la vista en l'apartat 3.4, el factor econòmic en principi no presentaria un problema ja que tot i que l'estudi econòmic present a l'apartat 4 demostra que els costos de la maqueta no serien baixos, aquets es podrien reduir significativament utilitzant components alternatius. El factor ubicació, del qual se'n parla a l'apartat 5, també seria un factor a tenir en compte a l'hora de decidir la viabilitat de l'estudi sobre una ubicació concreta. Pel que fa a les implicacions ambientals la maqueta en sí no presenta cap problema donades les seves dimensions i el fet de que no genera residus, però la futura instal·lació d'una planta d'energies renovables sí que presentaria implicacions ambientals. Aquestes es troben recollides a l'apartat 6.

Finalment es presenta un resum de com s'acabaria decidint si en una ubicació concreta és viable o no instal·lar una planta d'energies renovables, ja sigui solar o eòlica.

3.5.1 Viabilitat instal·lació solar

Un panell solar està format per diverses cel·les solars connectades en sèrie per proporcionar una tensió major [9].

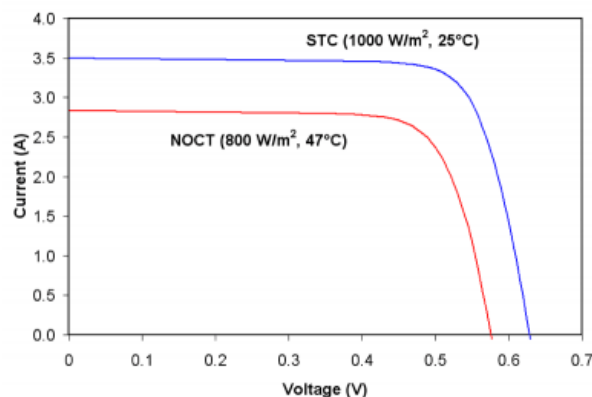


Figura 23: Gràfic Tensió - Intensitat cel·la solar

Com es pot observar (veure Figura 23), proporcionen una tensió variable depenent de l'energia solar que reben i una intensitat fixa. Per tant, mesurar la tensió rebuda serà suficient per poder estudiar el rendiment dels panells solars.

Cada cel·la, segons tipus i radiació solar produeix entre 0.35 i 0.65 V. Per exemple, un panell de 12 V conté 36 cel·les connectades en sèrie proporcionant una tensió nominal de 18 V, suficientment alta per carregar bateries de 12 V.

Un cop vist això, per calcular la viabilitat de la instal·lació de panells solars es procedirà a calcular la potència mitjà (P_{Real}) recollida pel sensor solar (a partir de la mitja de les tensions recollides ($V_{Mitjà}$) multiplicat per la intensitat (I) de funcionament) i es dividirà aquest valor entre la potència màxima ($P_{Màxima}$) que el sensor pot proporcionar per obtenir el rendiment mitjà (μ_{Panell}).

$$\mu_{Panell} = \frac{P_{Real}}{P_{Màxima}} = \frac{V_{Mitjà} * I}{P_{Màxima}}$$

Un rendiment òptim oscil·la entre el 75 i el 85%.

Un cop obtingudes les dades es pot comparar entre diferents ubicacions i decidir-ne la òptima.

3.5.2 Viabilitat instal·lació eòlica

Com es va comentar a l'apartat 3.1.1.1, el sensor eòlic escollit en aquesta maqueta no és l'ídoni per realitzar aquest estudi sobre viabilitat donat que la sortida en tensió que proporciona no està relacionada directament sobre la velocitat del vent que rep el sensor, i per acabar utilitzant aquest sensor seria necessari realitzar un estudi previ on es prenguessin dades i s'extragués una gràfica relacionant la tensió de sortida del sensor respecte la velocitat del vent.

Per altra banda, també es podria canviar aquest sensor per un anemòmetre comercial, preferiblement amb sortida de tensió el qual ens facilitaria molt l'adaptació del circuit reduint-la a un simple canvi en el divisor de tensió i l'escalat posterior per programa. Un exemple en podria ser l'anemòmetre ANEMO 4403 0-10V [10] (veure Figura 24), que com es requeria anteriorment proporciona una sortida de 0 a 10 V relacionats de manera proporcional a la velocitat del vent, com es pot observar a la següent gràfica (veure Figura 25).

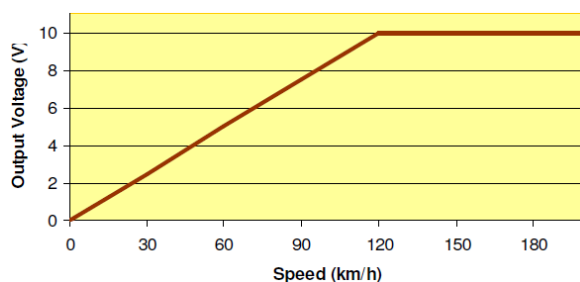


Figura 25: Gràfic senyal sortida ANEMO 4403.



Figura 24: ANEMO 4403.

El motiu d'haver escollit el sensor (Aerogenerador - WINDPITCH educational kit) és el de proporcionar a la maqueta major similitud a un possible aerogenerador, dotant la maqueta de major utilitat de cara a futurs estudis amb aquesta, com podria ser, l'estudi del rendiment obtingut segons el tipus d'aspes o la inclinació d'aquestes.

Un cop obtingudes les dades (velocitat del vent) i dibuixades, s'hauria de procedir a realitzar-ne un estudi per avaluar, entre altres factors, la velocitat mitjana i les fluctuacions. Com a referència [11], la velocitat mínima per a que un aerogenerador mig pugui funcionar és de 11km/h, aconseguint el seu màxim rendiment als 45 km/h i amb una velocitat màxima de funcionament de 100km/h (aquestes dades varien en funció del tipus d'aerogenerador).

La velocitat mitjana mínima anual per a obtenir un bon rendiment és de 13 km/h.

Per altra banda, els aerogeneradors connectats directament a una càrrega a diferència dels connectats a un generador, anomenats molins de vent, que usualment posseeixen moltes més pales, aconsegueixen aprofitar molt millor el vent a baixes velocitats. Aquests presenten una velocitat mínima d'arrencada de 4'8 km/h i presenten la característica de no poder augmentar la seva velocitat de rotació un cop la velocitat del vent supera els 28 km/h, el qual és molt beneficiós per a la utilitat que solen presentar la majoria de molins de vent, bombejar aigua, procés que requereix continuïtat.

4 ESTUDI ECONÒMIC

En aquest apartat es troba el pressupost detallat amb el corresponent desglossament d'amidaments dels components.

S'han escollit diferents distribuïdors en funció del preu en que ofertaven els productes requerits, com poden ser; Arduino, RS, Amazon, etc.

	Cablejat	Unitats	Amidaments	Nº	Secció	Preu/unitat	Total
1	Cable Unipolar flexible 1,5 mm ² GENLIS F 750V General Cable	m	10	1	1'5 mm	0,141 €	1,41 €
TOTAL CABLEJAT							1,41 €
	Sensors	Unitats	Amidaments	Nº		Preu/unitat	Total
1	Aerogenerador WINDPITCH educational kit	mm	250*400*15	1		132,98 €	132,98 €
2	Panell solar - SR5-36 Solar Panel	mm	194*385*28	1		59.14 €	59.14 €
TOTAL SENSORS							192,12 €
	Alimentació	Unitats	Amidaments	Nº		Preu/unitat	Total
1	Bateria d'àcid-plom de cicle intens 1'2Ah i 12V RS 537-5444	mm	58 x 97 x 43	1		10,71 €	10,71 €
TOTAL ALIMENTACIÓ							10,71 €

	Arduino	Unitats	Amidaments	Nº	Preu/unitat	Total
1	Arduino UNO	-	-	1	21,4 €	21,4 €
TOTAL ARDUINO						21,4 €
	Adaptadors	Unitats	Amidaments	Nº	Preu/unitat	Total
1	Regulador de càrrega SunGuard 4A 12V Morningstar	-	-	1	40,41 €	40,41 €
2	circuit - LM7805 (2 condensadors 100µF 1 fusible 1'5A 1 resistència 680Ω 1 LM7805)	-	-	1	4 € (aprox.)	4 €
3	Divisor de tensió (4 resistències)	-	-	4	0'05 €	0,2 €
TOTAL ADAPTADORS						44,61 €
	Connexió telefònica	Unitats	Amidaments	Nº	Preu/unitat	Total
1	Arduino GSM Shield	-	-	1	83,49 €	83,49 €
2	GSM Modem Wavecom Q2303A Module USB Interface AT Commands SMS 900/1800MHz	-	-	1	35,37 €	35,37 €
TOTAL CONNEXIÓ TELEFÒNICA						118,86 €
PRESSUPOST TOTAL						389,11 €

5 CONDICIONS PER A LA SEVA EXECUCIÓ

En aquest apartat es troben les condicions necessàries per a poder instal·lar la maqueta en una ubicació amb garantia de funcionament i un petit manual sobre la seva utilització.

5.1 Planificació ubicació sensors

Requisits necessaris per a la instal·lació de la maqueta amb garanties. Dividirem aquest apartat en quatre subapartats, tots quatre molt importants i a tenir en compte. També s'ha de tenir en compte que s'han d'evitar els obstacles al voltant de la maqueta que puguin provocar ombres sobre el panell solar o actuar de pantalla reduint la velocitat del vent. Per tant, el primer a vigilar és la situació dels sensors un respecte a l'altre perquè no s'afectin entre ells.

5.1.1 Cobertura GSM

El primer requisit per al correcte funcionament de la maqueta és disposar de cobertura GSM ja que és indispensable per a la seva comunicació amb el PC receptor. En concret la banda de 900MHz que és la utilitzada. Aquesta banda la utilitzen la majoria de teleoperadors per proporcionar cobertura 2g i algunes per 3g.

Com es pot observar en la següent imatge (veure Figura 26), la cobertura (en aquest cas 2g) avui en dia arriba gairebé a tot el territori Espanyol, però encara queden ubicacions on aquesta és nul·la o insuficient i és important tenir-ho en compte.



Figura 26: Cobertura 2g al territori Espanyol.

5.1.2 Mínim d'hores de sol (per alimentar la bateria)

Un altre punt important és el mínim d'hores de sol al dia o el màxim d'hores sense sol que pot suportar la maqueta per poder proporcionar un funcionament òptim, ja que si la bateria es queda descarregada no es pot alimentar l'arduino per seguir prenent dades ni el mòdem de comunicació.

Primerament es calcula el temps de càrrega de la bateria, es tindrà en compte un 10% de pèrdua d'eficiència respecte els 300 mA proporcionats pel panell solar.

$$\text{Temps de Càrrega} = \frac{\text{Capacitat de la bateria (mAh)}}{\text{Intensitat de càrrega (mA)} * 90\%} = \frac{1200}{270} = 4'4 \text{ hores}$$

Seguidament es calcula la intensitat consumida pels components que requereixen d'alimentació.

Pel que fa al mòdem (Arduino GSM Shield) requereix 1000 mA però només caldrà fer la emissió de dades un cop al dia i aquesta emissió triga de l'ordre de pocs segons depenent del temps que s'escull entre una presa de dades i la següent. Per fer uns càlculs aproximats, s'escullen 5 segons com a exemple.

$$\text{Temps de Descàrrega} = \frac{\text{Capacitat de la bateria (mAh)}}{\text{Intensitat de descàrrega (mA)}} = \frac{1200 \text{ mAh}}{1000 \text{ mA}} = 1'2 \text{ hores}$$

D'aquestes 1'2 hores només en farem servir 5 segons per tant la part proporcional són:

$$1'2 \text{ hores} = \frac{3600s}{1h} = 4320 s$$

$$\% \text{ utilització bateria} = \frac{5s}{4320s} * 100 = 0,115\%$$

Com es pot observar aquest percentatge és insignificant, per tant no és necessari tenir-lo en compte.

Per altra banda es troba l'arduino que presenta un consum aproximat de 40 mA. En aquest cas prendrem com a referència el pitjor dels casos, que l'arduino es trobi en constant funcionament, sense posar-lo en mode de baix consum en cap moment. Per tant, el temps de descàrrega serà de:

$$\text{Temps de Descàrrega} = \frac{\text{Capacitat de la bateria (mAh)}}{\text{Intensitat de descàrrega (mA)}} = \frac{1200 \text{ mAh}}{40 \text{ mA}} = \mathbf{30 \text{ hores}}$$

Aquest serà el màxim d'hores que la maqueta podrà suportar sense intensitat llumínica, ja que la bateria acabaria sense càrrega i es perdria l'alimentació de la maqueta inutilitzant-la fins a tornar a rebre llum solar. Tot i això, cal recordar que tot i fer núvol, existeix llum i aquesta continuaria proporcionant energia, per tant seria molt difícil esgotar la bateria donades les condicions solars de la nostra zona geogràfica.

L'altre punt mencionat inicialment és el mínim número d'hores solars al dia, que es calcula a continuació:

$$\text{Consum} = 40 \text{ mA} * 24 \text{ h} = 960 \text{ mA} * \text{dia}$$

Per tant, necessitem una capacitat total de 960 mA al dia.

$$\text{Hores Càrrega mínimes} * \text{dia} = \frac{\text{Capacitat mínima (mA} * \text{dia)}}{\text{Intensitat de càrrega (mA)} * 90\%} = \frac{960}{270} = \mathbf{3'5 \text{ hores}}$$

5.1.3 Mínima altura per l'aerogenerador

Un altre punt a tenir en compte és no situar la maqueta al terra, ja que qualsevol obstacle podria impedir que l'aerogenerador rebés el vent adequat com per a prendre dades objectives. Per tant s'haurà de situar la maqueta en un punt mitjanament elevat com pot ser la part superior d'una edificació, d'una petita estructura o el capdamunt d'una muntanya.

5.1.4 Inclinació i orientació placa solar

L'orientació de la placa ha de ser cap al sud geogràfic, per aprofitar al màxim l'energia solar.

La inclinació de la placa solar també influirà sobre les dades obtingudes per aquesta. A continuació es pot observar un breu resum sobre com calcular la inclinació òptima respecte el sol [12]. Tanmateix, la pròpia maqueta servirà per acabar decidint la inclinació òptima en una ubicació determinada.

Hi ha dos mètodes per a calcular la inclinació òptima anual;

$$\beta_{opt1} = 3,7 + 0,69 \cdot |\phi|$$

$$\beta_{opt2} = |\phi| - 10$$

I un bon mètode pot ser escollir la inclinació mitjana obtinguda a partir dels dos mètodes.

$$\beta_{opt Mitja} = (\beta_{opt1} + \beta_{opt2}) / 2$$

On;

$|\phi|$ = Latitud

β_{opt} = Inclinació òptima

5.2 Manual d'ús

La posada en marxa de la maqueta és molt senzilla. Primerament s'ha de carregar el programa (3.3.2) a l'arduino. A continuació, una vegada la maqueta estigui correctament instal·lada en la ubicació escollida i els sensors correctament situats seguint les permutes indicades en l'apartat 5.1, es pot procedir a iniciar el programa al PC (3.3.3). Un cop obert, s'ha de depurar el programa polsant sobre la següent icona (veure Figura 27):

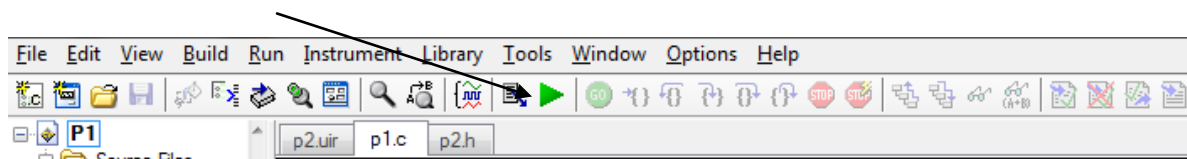


Figura 27: Inici Labwindows

A continuació el programa s'iniciarà i ens demanarà el directori on volem guardar el fitxer amb les dades i el nom del fitxer (veure Figura 28):

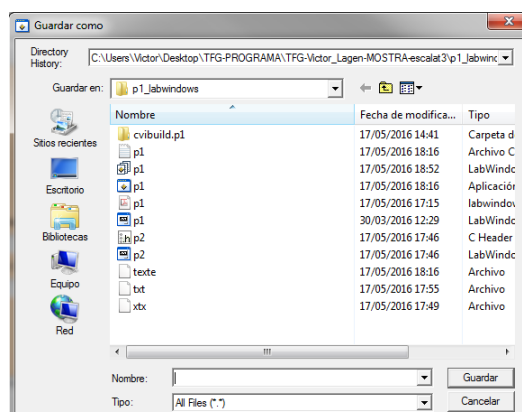


Figura 28: Directori i fitxer de dades



Posteriorment, s'obrirà el panell del programa (veure Figura 29):

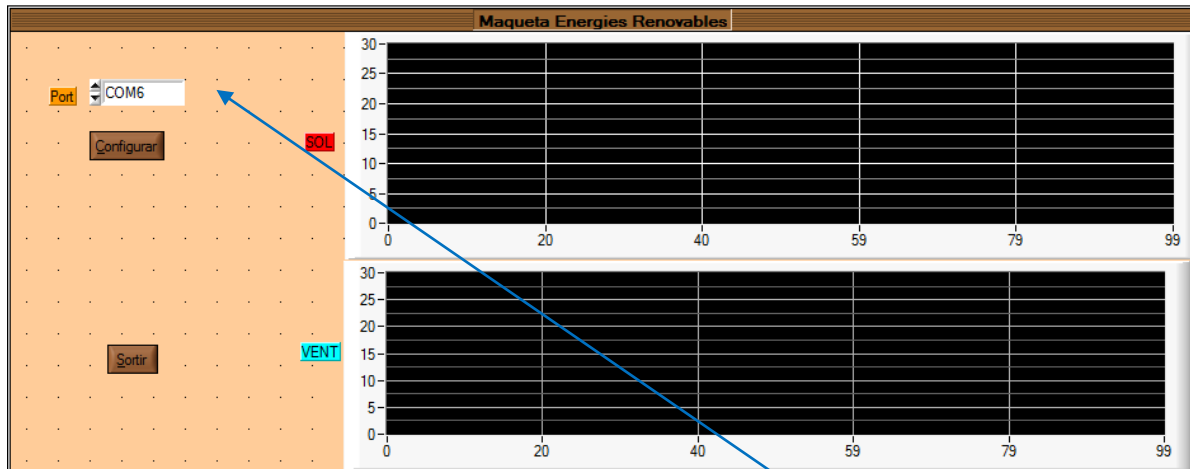


Figura 29: Panell Labwindows

I haurem de configurar el port del PC al qual hi tenim el port sèrie:

Finalment caldrà polsar sobre el botó Configurar.

Un cop realitzats aquets passos, el programa anirà recollint dades i emmagatzemant-les al fitxer seleccionat fins que el vulguem aturar, per fer-ho caldrà polsar sobre el botó Sortir.

Arribats a aquest punt, al fitxer hi trobarem les dades recollides, en un format com el de la següent imatge (veure Figura 30), adequat per exportar a Excel i realitzar la gràfica o extracció de variables necessària.

txt: Bloc de notas			
Archivo	Edición	Formato	Ver
temps= 0.0	13.64	20.55	
temps= 0.5	13.64	20.55	
temps= 1.0	13.64	20.55	
temps= 1.5	13.72	20.55	
temps= 2.0	13.64	18.05	
temps= 2.5	13.64	20.55	
temps= 3.0	13.64	20.55	
temps= 3.5	13.64	20.55	
temps= 4.0	13.64	20.55	
temps= 4.5	13.64	20.55	
temps= 5.0	13.72	20.55	
temps= 5.5	13.64	20.55	
temps= 6.0	13.64	20.55	
temps= 6.5	13.64	20.55	
temps= 7.0	13.64	20.55	
temps= 7.5	13.64	20.55	
temps= 8.0	13.64	20.55	
temps= 8.5	13.64	20.55	
temps= 9.0	13.64	20.55	
temps= 9.5	13.64	20.55	

Figura 30: Dades fitxer

6 IMPLICACIONS AMBIENTALS

6.1 Avantatges Energies Renovables

Es pot parlar dels avantatges de les energies renovables tenint en compte les limitacions d'aquesta simplificació. Els beneficis depenen del tipus d'energia i tecnologia utilitzada.

Tot i això, és evident que existeix una necessitat econòmica i ambiental urgent de diversificar la producció actual d'energia.

El principal benefici que aporten les energies renovables en termes ambientals és la reducció de les emissions de CO₂ i altres gasos contaminants. Amb això ajudem a reduir el canvi climàtic degut a factors humans. Les energies que no presenten component biològic en el seu procés d'obtenció afavoreixen aquest objectiu.

Les energies renovables, a part de gasos, també redueixen la generació de residus de tot tipus o ajuden a eliminar i transformar certs subproductes industrials o biològics (com és el cas del biogàs).

El fet de que es tracta d'energies inesgotables també és un factor decisiu de cara al futur.

Les millores tecnològiques permeten el continu desenvolupament i millora de les fonts d'obtenció d'aquestes tecnologies, fent-les més productives i augmentant el seu rendiment. Energies netes com la fusió nuclear (que es troba avui en dia en els seus inicis) poden suposar un gran pas en termes energètics de cara a un futur proper.

6.2 Inconvenients Energies Renovables

Com en l'apartat anterior parlarem en termes generals sobre els inconvenients de les energies renovables, també a tenir en compte.

Un dels principals és la ocupació de grans extensions de sòl. L'energia solar, eòlica i la producció de biocombustibles competeixen per aquest escàs recurs, que té com a ús alternatiu la producció d'aliments o el manteniment dels ecosistemes naturals. La elecció d'entorns amb escàs potencial biològic pot minimitzar-ne el problema. Altres tipus d'energies (biogàs, hidràulica, biomassa, etc.) no presenten aquest problema.

També ens trobem el dilema ètic de enfrontar producció d'aliments i d'energia pel que fa als biocombustibles. O la utilització de abonaments i fertilitzants per augmentar la productivitat de la terra.

Un altre inconvenient és l'impacte sobre el paisatge en el medi rural, especialment motivat per l'acumulació d'instal·lacions. Les més agressives són les eòliques i solars, donades les característiques pròpies i els espais òptims on s'instal·len.

També presenten inconvenients les energies que presenten factor biològic com el cas dels biocombustibles, generen CO₂.

La resta de inconvenients no són rellevants, ja que poden ser solucionats mitjançant la tecnologia o amb mesures preventives o correctores de fàcil implantació.

6.2.1 Impacte ambiental energia eòlica

La generació d'energia eòlica prové de l'aprofitament del vent, que és un efecte indirecte de l'energia que genera el sol.

L'energia que arriba a la Terra no es distribueix uniformement a causa de les diferents característiques de les superfícies sobre les que incideix (especialment masses d'aigua i continents) i de la diferent inclinació respecte al sol dels diferents punts de l'esfera terrestre. Això origina unes diferències de temperatures en l'atmosfera, que al tendir a igualar-se generen corrents d'aire.

Aquest efecte ha estat aprofitat utilitzant una tecnologia mecànica des de temps pretèrits per moure l'aigua. Actualment la tecnologia utilitzada per aprofitar l'energia del vent, transforma l'energia mecànica en elèctrica permetent qualsevol ús de la mateixa.

Els parcs eòlics van començar a estendre's en l'últim quart del segle XX pels països industrialitzats com a alternativa a la dependència energètica del petroli.

Actualment aquest factor i les implicacions dels combustibles fòssils en el canvi climàtic han potenciat tant la instal·lació de parcs com les inversions en tecnologia.

S'han de considerar en aquest tipus d'instal·lacions els impactes provocats pels molins, les estructures de manteniment, equip elèctric associat i per les línies elèctriques.

6.2.1.1 Afectació al paisatge

El paisatge és el factor ambiental que més incideix sobre el medi ambient en aquest tipus d'energia. Les mides dels molins i les pales van en augment, incrementant la seva visibilitat. I la posició estratègica idònia d'instal·lació dels parcs eòlics usualment coincideix amb zones d'alta visibilitat.

L'impacte provocat per la línia elèctrica que connecta el parc és irrellevant comparat amb els efectes del parc sobre el paisatge.

6.2.1.2 Afectació a la flora

S'ha d'escollir localitzacions que no siguin espais naturals d'especial interès per a instal·lar els parcs. Procurant que no presentin espècies endèmiques o protegides.

És preferible escollir espais alternatius a implantar costoses mesures correctores.

6.2.1.3 Afectació a la fauna

L'impacte més rellevant sobre la fauna d'aquests parcs es produeix sobre les aus. És molt important evitar rutes migratòries o espais on hi existeixin espècies protegides.

Estudis previs sobre la mortalitat que provoquen els parcs demostren que la majoria de casos es concentren en un número molt reduït d'aerogeneradors. Per tant, modificant la localització d'aquests redueix notablement aquest efecte sobre les aus.

Les línies elèctriques que connecten els parcs solen provocar en molts casos major mortalitat que els propis parcs.

6.2.1.4 Afectació en soroll

També és important tenir en compte el factor soroll a l'hora d'escollir la localització d'un parc ja que pot ocasionar molèsties en poblacions properes.

Estudis previs permeten localitzar-los en punts on aquest factor sigui irrellevant.

6.2.1.5 Afectació sobre el sòl

L'impacte sobre l'estructura i manteniment dels sòls és un factor pràcticament irrellevant sempre i quan un cop finalitzada la construcció, es prenguin les mesures necessàries per mantenir la cobertura vegetal dels terrenys. Aquesta cobertura evita que en parcs situats sobre pendent es produeixin pèrdues importants del sòl i en limita l'erosió.

6.2.2 Impacte ambiental energia Solar Fotovoltaica

L'energia solar fotovoltaica s'obté a partir de la conversió de la radiació solar en energia de corrent continua mitjançant panells solars formats per materials semiconductors. Després es transforma en corrent alterna mitjançant alternadors i posteriorment a través de transformadors s'allibera a la xarxa elèctrica per a la seva distribució.

6.2.2.1 Afectació sobre el paisatge

El paisatge és el factor ambiental més important a tenir en compte en aquest tipus d'energia. Les plaques solars requereixen de certa construcció i són difícilment integrables a l'entorn. També s'ha de tenir en compte el reflex que aquests parcs produeixen, visibles a grans distàncies.

Per atenuar aquesta afectació, es poden situar aquests parcs sobre conques visuals del paisatge. Es poden incorporar elements del paisatge per actuar com a pantalla correctora, preferiblement vegetació autòctona.

També s'han de tenir en compte els elements auxiliars a aquestes plantes, com els transformadors i les línies elèctriques associades que són estructures grans i proporcionen un impacte paisatgístic important.

6.2.2.2 Afectació sobre el sòl

Es important considerar que el terreny ocupat per aquets parcs, es terreny potencialment agrícola, per tant reduïm la producció alimentària generant un dilema energètic - alimentari. S'han d'afavorir sempre zones amb menor capacitat biològica.

L'ocupació del sòl motiva la desestructuració d'aquest i la pèrdua de cobertura vegetal, afavorint-ne l'erosió i en conseqüència la pèrdua de sòl. És aconsellable realitzar un manteniment adequat per evitar aquest potencial problema.

6.2.2.3 Afectació sobre la flora

S'ha d'evitar implantar parcs solars en espais naturals d'especial interès, evitant també la presència d'espècies endèmiques o protegides. Estudis de vegetació són necessaris per definir la idoneïtat de la zona.

6.2.2.4 Afectació sobre la fauna

Els parcs solars no presenten un impacte massa apreciable sobre la fauna, tot i que s'han de tenir en compte alguns factors. Per exemple, les espècies que nidifiquen a superfície com n'és l'exemple la avutarda.

Afectacions com les produïdes al sistema hidrològic són fàcilment corregibles. Les estructures necessàries per instal·lar els panells no requereixen de gaire profunditat, el qual no ens genera problemes d'aquesta índole.

7 LEGISLACIÓ ESPANYOLA RESPECTE LES ENERGIES RENOVABLES

Per entendre la situació actual de les energies renovables al nostre país no només és necessari endinsar-se en la tecnologia actual referent a la producció d'aquestes, la qual es troba molt avançada, ni en les condicions climàtiques que es donen en el territori, també influeix de manera decisiva en la implantació d'aquests tipus d'instal·lacions la legislació vigent.

La legislació, o regulació espanyola de les energies renovables [13] actualment no afavoreix ni molt menys les energies renovables, les quals van experimentar el seu màxim esplendor amb l'antiga regulació (1998 - 2013) quan es van crear moltes instal·lacions d'aquest caire sobre el territori espanyol. Instal·lacions que han sofert una forta patacada a partir de la regulació actual (2013 - 2020). El decret presentat pel Ministre d'Indústria, José Manuel Soria, suposa una retallada en incentius de 1.370 milions d'euros en energies renovables [14], el que significa una baixada del 15% per a l'any 2014 i posteriori respecte el pressupost del 2013.

Això comporta que l'eufòria produïda per la legislació aprovada el 1998 respecte les energies renovables es tradueixi a dia d'avui en un estancament massiu en termes de creixement d'aquestes energies, restant a l'espera d'una nova regulació que fomenti les energies renovables en el nostre país i el situï a nivell dels primers productors europeus.

8 CONCLUSIONS

Tal i com es va comentar a la introducció, l'objectiu general d'aquest projecte és el disseny d'una maqueta capaç d'obtenir informació de dues de les fonts energètiques renovables més importants; el sol i el vent, per tal de permetre decidir la viabilitat de la instal·lació d'equipament energètic renovable en una ubicació concreta.

El projecte consta de dues parts, la primera i més important és l'estudi pràctic portat a terme, on s'ha dissenyat la maqueta encarregada de l'obtenció de dades i se li ha proporcionat el programa necessari per portar a terme aquesta tasca. I una segona part teòrica, on s'estableixen i es calculen els requisits necessaris per dotar la maqueta de total autonomia tant pel que fa referència a l'alimentació com a la connexió sense fils i les modificacions al programa per tal d'optimitzar el consum de la maqueta.

Pel que fa a la primera part, el resultat ha estat una maqueta formada per dos sensors, una placa solar i un petit aerogenerador, els quals al proporcionar una tensió de sortida en funció de l'energia rebuda (ja sigui solar o eòlica) permeten recollir dades a l'Arduino, un cop adaptats els nivells de tensió. El programa està calculat per recollir un màxim de 512 dades de cada sensor al dia, formant un total de 1024 dades al dia, valor que ve limitat per la capacitat de la EEPROM de l'Arduino (1kb). Per fer-ho es recull una dada de cada sensor cada 170 segons. El motiu de fer servir aquesta memòria ve donat per la capacitat de no perdre les dades en cas de pèrdua d'alimentació, possibilitat que apareix en l'estudi teòric. Seguidament, l'Arduino s'encarrega d'enviar totes les dades al PC receptor el qual realitza una petició cada 86.400 segons (1 dia) i tracta aquesta informació convenientment per poder ser analitzada.

S'ha realitzat un estudi sobre el terreny (veure 3.2.1) al parc eòlic de Rubió utilitzant un programa de 30 minuts de durada. Les conclusions extretes a partir dels valors obtinguts indiquen que la placa solar proporciona una tensió elevada i força constant, amb la qual es pot calcular la potència mitja fàcilment donat que la intensitat que aquesta proporciona és constant, per posteriorment calcular el rendiment sobre la potència màxima. De tota manera, seria necessari un estudi amb un llarg mostreig per proporcionar major fiabilitat. Pel que fa a l'aerogenerador els resultats obtinguts no són tant satisfactoris, proporciona valors molt variables, tot i que això no n'és el principal inconvenient.

Com es comenta a la descripció de l'aerogenerador (veure 3.1.1.1), es va escollir aquest sensor per proporcionar major aparença a una maqueta "real" i amb la perspectiva de futurs estudis com pot ser l'eficiència de l'aerogenerador en funció del tipus de pales o de la inclinació d'aquestes. Sense tenir en compte que no existeix cap relació entre la tensió que aquest proporciona i la velocitat del vent que se li aplica, impossibilitant el càlcul de la velocitat mitja del vent o la distribució de velocitats al llarg del temps, valors necessaris per a decidir la viabilitat de tal energia. El sensor òptim seria un anemòmetre, per exemple el ANEMO 4403 (veure 3.5.2), però en cas de voler continuar amb el petit aerogenerador, caldria realitzar un estudi que relacionés la tensió de sortida d'aquest amb la velocitat de vent rebuda, per tal de poder obtenir dades reals contrastades.

El programari utilitzat, tant el processing de l'arduino com Labwindows ha proporcionat molt bon resultat. Tot i les limitacions que pot tenir un llenguatge com el processing de l'arduino, la flexibilitat que proporciona en matèria d'aprenentatge i la gran quantitat de llibreries a l'abast de l'usuari el converteixen en un llenguatge molt atractiu per a la confecció d'una maqueta.

A la segona part, l'estudi teòric, per tal de dotar d'autonomia a la maqueta s'ha escollit una bateria de 1'2 Ah a 12 V, calculada per suportar 30 hores de descàrrega (veure 5.1.2), que es carrega a partir del panell solar, el qual a part de funcionar com a sensor també s'utilitza com a generador i proporciona energia a la bateria a través d'un regulador comercial. La bateria és l'encarregada de proporcionar autonomia energètica tant a l'arduino com al mòdem encarregat de la comunicació sense fils de la maqueta amb el PC receptor. Aquesta alimentació es realitza a través d'un circuit adaptador de tensió utilitzant un LM7805 per ajustar-la a 5 V. El mòdem escollit és el Arduino GSM Shield, el qual es comunicarà amb un altre mòdem (GSM Modem Wavecom Q2303A) instal·lat al PC, mitjançant una freqüència GSM900MHz i un baud rate de 115.200 bits/s (el màxim acceptat per la comunicació serial de l'Arduino).

Les modificacions teòriques aportades al programa consisteixen en introduir l'Arduino en mode de baix consum, per fer-ho possible cal afegir una interrupció a partir de l'arribada de dades pel port sèrie, el qual és possible connectant el Pin 2 amb el Rx aconseguint així fer baixar 5V pel Pin 2 a l'arribada de dades pel serial, disparant la interrupció (INT 2). Amb això s'aconsegueix que el programa funcioni únicament per interrupcions, ja que la recollida de dades dels sensors es fa a partir del TimerOne, llibreria que programa una interrupció interna. El mode de baix consum escollit és el SLEEP_MODE_PWR_SAVE. Això suposaria un estalvi molt considerable d'energia ja que es reduiria el consum de l'Arduino de 40 mA a 1'64 mA durant gran part del temps, donat que aquest només sortiria del mode de baix consum una vegada cada 170 segons per recollir dades i una vegada al dia per enviar-les pel serial. Allargant així el temps de descàrrega de la bateria.

No obstant, tot i no implementar aquesta millora, el temps de descàrrega de 30 hores de la bateria és molt difícil d'esgotar donades les condicions climàtiques de la ubicació geogràfica catalana, i més, tenint en compte que en cas de suportar condicions climàtiques adverses, la intensitat lumínica segueix sent captada per la placa solar, proporcionant energia tot i no ser un dia assolellat.

Sense perdre de vista l'objectiu i un cop obtinguda la informació, el càlcul de la viabilitat d'energies renovables és molt senzill, referent a l'energia fotovoltaica el rendiment òptim oscil·la entre el 75% i el 85%. Un cop estudiades diferents ubicacions seria suficient amb contrastar les dades d'unes i altres per decidir-ne la més adient. Per altra banda, l'energia eòlica aconsegueix ser eficient a partir dels 13 Km/h de velocitat del vent, amb un rang de funcionament entre 11 i 100 km/h i amb una producció màxima als 45 Km/h, per tant seria necessari analitzar les dades obtingudes per exemple mitjançant gràfiques de valors entre límits per poder contrastar unes dades amb altres objectivament.

Tenint en compte l'increment en estalvi energètic que la maqueta pot aconseguir en el cas d'instal·lar equipament energètic, el pressupost necessari per construir-la no és excessiu. Tanmateix un bon estudi previ referent a la planificació en la ubicació dels sensors és imprescindible per assegurar un rendiment òptim en la recollida de dades.

Finalment, és important remarcar que la legislació vigent no és gens favorable envers les energies renovables des del decret presentat pel Ministre d'Indústria l'any 2013. Això ha desmotivats el mercat a invertir en aquest sector tot i que s'espera que la legislació s'ajusti al marc europeu i torni a apostar per les energies renovables.

9 BIBLIOGRAFIA

- [1] Fadisel. (2016). Electrónica, iluminación, energía solar y kits educativos. [online]
Available at: http://fadisel.es/educatiu-cebekit/aerogenerador-en-kit/wind-pitch-education-kit-c-7117-c-7117_r_455_2418.aspx [Accessed 1 Jun. 2016].

- [2] Es.farnell.com. (2016). *SR5-36 - RALOSS - PANEL SOLAR, 5W | Farnell element14 España*. [online] Available at: <http://es.farnell.com/raloss/sr5-36/panel-solar-5w/dp/1462491?selectedCategoryId=&exaMfpn=true&categoryId=&searchRef=SearchLookAhead> [Accessed 1 Jun. 2016].

- [3] Es.rs-online.com. (2016). *| Batería de ácido-plomo de ciclo intenso, RS Pro, 1.2Ah |*. [online] Available at: <http://es.rs-online.com/web/p/baterias-recargables-de-plomo-acido/5375444/> [Accessed 1 Jun. 2016].

- [4] Autosolar.es. (2016). Regulador de carga Morningstar. [online] Available at: <https://autosolar.es/pdf/regulador%20de%20carga%204,5%20amperios%2012V%20morningstar.pdf> [Accessed 1 Jun. 2016].

- [5] Tuelectronica.es. (2016). *Regulador de tensión a 5v - TuElectronica.es*. [online] Available at: <http://www.tuelectronica.es/tutoriales/electronica/regulador-de-tension-a-5v.html> [Accessed 1 Jun. 2016].

- [6] Arduino.cc. (2016). *Arduino - ArduinoGSMShield*. [online] Available at: <https://www.arduino.cc/en/Main/ArduinoGSMShield> [Accessed 1 Jun. 2016].

- [7] Amazon.com. (2016). *Amazon.com: Generic GSM Modem with Wavecom Q2303A Module USB Interface AT Commands SMS: Computers & Accessories*. [online] Available at: <http://www.amazon.com/Generic-Wavecom-Q2303A-Interface-Commands/dp/B00BD0V57I> [Accessed 1 Jun. 2016].

- [8] Arduino.cc. (2016). *Arduino - Home*. [online] Available at: <https://www.arduino.cc/> [Accessed 1 Jun. 2016].

- [9] Berg, H. (2016). *Paneles Solares de Silicio - Energía solar y eólica en Peru*. [online] Deltavolt.pe. Available at: <http://deltavolt.pe/energia-renovable/energia-solar/panel-solar-cristalino> [Accessed 1 Jun. 2016].

- [10] GmbH, P. (2016). *Anemómetro ANEMO 4403 0-10V | PCE Instruments*. [online] Pce-instruments.com. Available at: https://www.pce-instruments.com/espanol/instrumento-medida/medidor/anemometro-ied-electronics-anem_metro-anemo-4403-0-10v-det_255953.htm?_list=k&_listpos=17 [Accessed 1 Jun. 2016].

- [11] Soliclima.com. (2016). *Energía eólica, la energía del viento*. [online] Available at: <http://www.soliclima.com/eolica.html> [Accessed 1 Jun. 2016].

- [12] Ingelibre. (2014). *Inclinación óptima de placas fotovoltaicas*. [online] Available at: <https://ingelibreblog.wordpress.com/2014/01/09/inclinacion-optima-de-placas-fotovoltaicas/> [Accessed 1 Jun. 2016].

- [13] Energiaysociedad.es. (2016). *3.5. Regulación espanyola de las energías renovables*. [online] Available at: http://www.energiaysociedad.es/ficha/3-5-regulacion-espanola-de-las-energias-renovables#_ftn130 [Accessed 1 Jun. 2016].

- [14] Eleconomista.es. (2016). *España aprueba finalmente la nueva regulación de las energías renovables - elEconomista.es*. [online] Available at: <http://www.eleconomista.es/mercado-continuo/noticias/5841228/06/14/Espana-aprueba-finalmente-la-nueva-regulacion-de-las-energias-renovables.html> [Accessed 1 Jun. 2016].

- [15] Santiapps - Arduino (IoT), iOS, Android & Glass. (2015). *Arduino (IoT): Simple Tutorial Arduino Dormiente: Parte 1*. [online] Available at: <https://quique123.wordpress.com/2015/11/27/arduino-iot-simple-tutorial-arduino-durmiente-parte-1/> [Accessed 1 Jun. 2016].